

New Challenges in Post-Quantum Cryptography

MSCA DN-ID QUARC Consortium



AAU OPEN

AALBORG UNIVERSITY
OPEN PUBLISHING

New Challenges in Post-Quantum Cryptography

MSCA DN-ID QUARC Consortium

Core QUARC Authors

Abraham Cano Aguilera

Raphael Frantz

Dimosthenis Iliadis-Apostolidis

José Luis Imaña

Sokol Kosta

Daniel Christian Lawo

Joseph Meyer

Michał Podleś

Jerónimo Sánchez García

Idelfonso Tafur Monroy

Juan José Vegas Olmos

All Core QUARC Authors contributed to the preparation of this book and are listed in alphabetical order by surname.

External Contributors

Rana Abu Bakar, Faris Alhamed, Xavier Arnal i Clemente,

Filippo Cugini, Simon Rommel, Carlos Rubio Garcia, Catalina Stan

All external contributors are listed in alphabetical order by surname.

New Challenges in Post-Quantum Cryptography

By Abraham Cano Aguilera, Raphael Frantz, Dimosthenis Iliadis-Apostolidis, José Luis Imaña, Sokol Kosta, Daniel Christian Lawo, Joseph Meyer, Michał Podleś, Jerónimo Sánchez García, Idelfonso Tafur Monroy & Juan José Vegas Olmos

1st Edition 2026

© Authors 2026

Manuscript preparation, technical writing, synthesis, and LaTeX development: Athanasios Panagiotis Pelekoudas, Aalborg University

Cover design and graphic layout: Emilia Castillo Frederiksen, Aalborg University

ISBN: 97887-7642-316-2

DOI: 10.54337/aau.quarc

Published by: Aalborg University Open Publishing | www.open.aau.dk

This publication is published exclusively in electronic format and is only available online

Licence

Except where otherwise noted, this work is licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0).

Third-party material, including figures, tables, and other content reproduced or adapted from previously published sources, is subject to the credit lines, licences, and permissions indicated with the respective material. Such third-party material is not covered by the book's CC BY 4.0 licence unless explicitly stated otherwise.

Source and Attribution

The QUARC Project

This book has been developed in the context of QUARC (*Quantum Resistant Communications: Advance Learning in Applied PQC Training*), a Horizon Europe Marie Skłodowska-Curie Actions Doctoral Network Industrial Doctorate. Running from 2022 to 2026, QUARC brings together academic and industrial expertise to advance the practical deployment of post-quantum cryptography and to train researchers in quantum-resistant communication technologies. The project addresses the implementation and integration of post-quantum cryptography across secure communication systems, high-performance networking, and hardware and software platforms.

QUARC is funded by the European Union’s Horizon Europe research and innovation programme under the Marie Skłodowska-Curie Actions, Grant Agreement No. 101073355.

Partner Institutions

The QUARC consortium brings together three academic institutions: Eindhoven University of Technology (TU/e) in the Netherlands, Aalborg University (AAU) in Denmark, and Complutense University of Madrid (UCM) in Spain, together with NVIDIA as an industrial partner. The consortium provides a joint academic and industrial environment for the doctoral researchers involved in the project, with research, training, and collaboration taking place across the participating institutions.

Further information about the project and its partners is available at the project’s website¹ and at the European Commission’s public database CORDIS².

¹<https://quarc.website/>

²<https://cordis.europa.eu/project/id/101073355>

Core QUARC Authors

Abraham Cano Aguilera received his BSc in Telecommunications Technologies and Services Engineering and his MSc in Advanced Mathematics and Mathematical Engineering from Universitat Politècnica de Catalunya (UPC), Spain. He completed an industrial PhD at Eindhoven University of Technology in collaboration with NVIDIA within the QUARC project. His research focuses on applied and post-quantum cryptography, particularly lattice-based cryptography, high-speed secure communication, and the implementation and offloading of cryptographic protocols using Data Processing Units and other network accelerators.

Raphael Frantz received his BSc in Computer Science and MSc in High-Performance Visual Computing from Université de Reims Champagne-Ardenne, France. Within QUARC, his doctoral research combines academic work at Eindhoven University of Technology with research activities carried out in collaboration with NVIDIA. His research focuses on high-performance networking and programmable network accelerators, including Data Processing Units and SmartNICs, with particular interest in offloading communication and data-processing tasks from host systems.

Dimosthenis Iliadis-Apostolidis received his MEng in Electrical and Computer Engineering from Aristotle University of Thessaloniki, Greece. He conducted his doctoral research at Aalborg University within the QUARC project, including an industrial secondment with NVIDIA. His research focuses on programmable and secure data-center networking, including P4-based packet processing, Data Processing Units, network virtualization, network slicing, and the integration of post-quantum cryptography with IPsec.

José Luis Imaña received his MSc and PhD degrees in Physics from Complutense University of Madrid, Spain. He is an Associate Professor in the Department of Computer Architecture and Automation at Complutense University of Madrid. His research interests include computer arithmetic, finite-field computation, cryptographic hardware, reconfigurable architectures, and hardware implementation of post-quantum cryptography.

Sokol Kosta received his BSc, MSc, and PhD degrees in Computer Science from Sapienza University of Rome, Italy. He is an Associate Professor in the Professor Promotion Programme at the Department of Electronic Systems, Aalborg University. He has been involved in multiple EU and national projects as PI and coordinator, having graduated 5 PhD students so far. His research has been recognized internationally with 10+ awards, including the 2024 IEEE Infocom Test of Time award. In QUARC, he is the main supervisor of two Doctoral Candidates (DCs). His research interests include edge computing, networking, AI acceleration, mobile cloud computing, distributed systems, and cybersecurity of critical infrastructure.

Daniel Christian Lawo received his BSc and MSc degrees in Electrical Engineering and Information Technology from the Technical University of Munich, Germany. His doctoral

work combined research at Eindhoven University of Technology with an industrial research period at NVIDIA within the QUARC project. His research focuses on post-quantum cryptography for resilient communication systems, with particular emphasis on data-center networking, hardware offloading, and the integration of post-quantum mechanisms into protocols such as IPsec and MACsec.

Joseph Meyer received his undergraduate and graduate degrees in Electrical and Electronics Engineering from Tel Aviv University, Israel. He is an industrial doctoral researcher at Eindhoven University of Technology within the QUARC project, including an industrial secondment with NVIDIA. His research focuses on post-quantum cryptography and its acceleration on heterogeneous computing and Data Processing Unit architectures, including the offloading of post-quantum key-establishment mechanisms.

Michał Podleś received his MSc in Computer Science, specializing in Artificial Intelligence and Data Analysis, from AGH University of Science and Technology in Kraków, Poland. He is a doctoral researcher at Eindhoven University of Technology within the QUARC project, including an industrial secondment with NVIDIA. His work focuses on high-performance networking and DPUs, including network acceleration, memory-system performance, and the identification of architectural bottlenecks in high-speed computing systems.

Jerónimo Sánchez García received his BSc in Computer Science from Universidad de Almería and his MSc in High Performance Computing from Universidade da Coruña, Spain. He is a doctoral researcher at Aalborg University within the QUARC project, including an industrial secondment with NVIDIA. His research focuses on high-performance networking and communication offloading, including Data Processing Units, programmable network accelerators, and message-processing mechanisms for high-performance computing systems.

Idelfonso Tafur Monroy received his PhD in Electrical Engineering from Eindhoven University of Technology. He is a Full Professor in the Department of Electrical Engineering at Eindhoven University of Technology, where he leads research in quantum and terahertz systems. His research spans photonic and terahertz communication systems, integrated photonics, quantum communication, and technologies for high-capacity and secure communication infrastructures.

Juan José Vegas Olmos received degrees in Telecommunications and Electronic Engineering and obtained his PhD from Eindhoven University of Technology. He is Research Program Coordinator at NVIDIA in the area of Software Architecture for networking and is also an Adjunct Professor at Aalborg University. His research and industrial activities focus on high-performance communication systems, data-center networking, network acceleration, and the architecture of emerging secure communication technologies.

Source Material and Attribution

This book draws on research conducted within the QUARC project and on related work by members and collaborators of the consortium. A substantial part of the material has been synthesized from the doctoral theses of Abraham Cano Aguilera [1] and Daniel Christian Lawo [2], together with the following research publications and technical contributions:

- A. Cano Aguilera, X. Arnal i Clemente, D. C. Lawo, I. Tafur Monroy, and J. J. Vegas Olmos. “First End-to-End PQC Protected DPU-to-DPU Communications”. In: *Electronics Letters* 59.17 (2023), e12901. DOI: 10.1049/e112.12901
- D. Lawo, R. Frantz, A. Cano Aguilera, I. Tafur Monroy, and J. J. Vegas Olmos. “Implementation and Comparison of Dilithium/Kyber and Falcon/Kyber PQC Software Stack on Data Processing Units”. In: *Proceedings of the 27th Annual Symposium of the IEEE Photonics Benelux Chapter*. 27th Annual Symposium of the IEEE Photonics Benelux Chapter, IEEE Benelux 2023. Conference date: 23–24 November 2023. Ghent, Belgium, Nov. 2023. URL: <https://www.aanmelder.nl/ieee-ps-benelux-2023>
- A. Cano Aguilera, C. Rubio García, D. Lawo, R. Frantz, I. Tafur Monroy, J. Imaña, and J. Vegas Olmos. “Post-Quantum Secure Protocols for Communication Between Data Processing Units”. In: *Proceedings of IEEE Photonics Benelux Chapter Annual Symposium 2023*. 27th Annual Symposium of the IEEE Photonics Benelux Chapter, IEEE Benelux 2023. Conference date: 23–24 November 2023. Nov. 2023. URL: <https://www.aanmelder.nl/ieee-ps-benelux-2023>
- D. Lawo, R. Frantz, A. Cano Aguilera, X. Arnal I Clemente, M. Podleś, J. L. Imaña, I. Tafur Monroy, and J. J. Vegas Olmos. “Falcon/Kyber and Dilithium/Kyber Network Stack on NVIDIA’s Data Processing Unit Platform”. In: *IEEE Access* 12 (Mar. 2024), pp. 38048–38056. DOI: 10.1109/ACCESS.2024.3374629
- A. Cano Aguilera, R. Abu Bakar, F. Alhamed, C. Rubio Garcia, J. L. Imaña, I. Tafur Monroy, F. Cugini, and J. J. Vegas Olmos. “First Line-Rate End-to-End Post-Quantum Encrypted Optical Fiber Link Using Data Processing Units (DPUs)”. In: *Optical Fiber Communication Conference (OFC) 2024*. Optica Publishing Group, 2024, M1G.4. DOI: 10.1364/OFC.2024.M1G.4
- C. Rubio Garcia, A. Cano, J. J. Vegas Olmos, S. Rommel, and I. Tafur Monroy. “Integrating Quantum Key Distribution into TLS 1.3: A Transport Layer Approach to Quantum-Resistant Communications in Optical Networks”. In: *Optical Fiber Communication Conference (OFC) 2024*. Optica Publishing Group, 2024, Th3B.3. DOI: 10.1364/OFC.2024.Th3B.3
- A. Cano Aguilera, C. Rubio Garcia, D. Lawo, J. L. Imaña, I. Tafur Monroy, and J. J. Vegas Olmos. “In-line rate encrypted links using pre-shared post-quantum keys and DPUs”. In: *Scientific Reports* 14.1 (Sept. 2024), p. 21227. DOI: 10.1038/s41598-024-71861-x
- D. C. Lawo, R. Abu Bakar, A. Cano Aguilera, F. Cugini, J. L. Imaña, I. Tafur Monroy, and J. J. Vegas Olmos. “Wireless and Fiber-Based Post-Quantum-Cryptography-Secured IPsec Tunnel”. In: *Future Internet* 16.8 (2024), p. 300. DOI: 10.3390/fi16080300
- A. Cano, C. R. Garcia, R. Frantz, I. T. Monroy, J. L. Imaña, and J. J. Vegas. “Integrating Post-Quantum Cryptography Plugins for IPsec Offloads to Data Processing Units in the

- Cloud-Edge Continuum”. In: *2024 IEEE 32nd International Conference on Network Protocols (ICNP)*. IEEE, 2024, pp. 1–6. DOI: 10.1109/ICNP61940.2024.10858568
- D. Lawo, M. Podleś, R. Frantz, A. Cano Aguilera, D. Iliadis-Apostolidis, J. Sánchez García, S. Kosta, I. Tafur Monroy, J. L. Imaña, and J. J. Vegas Olmos. “The Zero-Tax Data Center: A Use Case Through Quantum Resilient Communications”. In: *2024 24th International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2024, pp. 1–4. DOI: 10.1109/ICTON62926.2024.10647337
 - C. Rubio García, A. Cano Aguilera, C. Stan, J. J. Vegas Olmos, S. Rommel, and I. Tafur Monroy. “Enhanced Network Security Protocols for the Quantum Era: Combining Classical and Post-Quantum Cryptography, and Quantum Key Distribution”. In: *IEEE Journal on Selected Areas in Communications* 43.8 (2025), pp. 2765–2781. DOI: 10.1109/JSAC.2025.3568011
 - A. Cano Aguilera, C. Rubio Garcia, D. C. Lawo, I. Tafur, J. L. Imaña, and J. J. Vegas Olmos. “First Demonstration of 200 Gbps Regime Line-Rate Quantum-Secure MACsec Optical Links Using Commodity Hardware Offloads”. In: *Optical Fiber Communication Conference (OFC) 2025*. Optica Publishing Group, 2025, Tu2D.2. DOI: 10.1364/OFC.2025.Tu2D.2
 - A. Cano Aguilera, I. Tafur Monroy, J. J. Vegas Olmos, and J. L. Imaña. “ARM Architecture Optimizations for Line-Rate PQC Communications”. In: *2025 International Conference on Optical Network Design and Modeling (ONDM)*. IEEE, 2025, pp. 1–4. DOI: 10.23919/ONDM65745.2025.11029331
 - A. Cano, J. L. Imaña, I. T. Monroy, and J. J. Vegas Olmos. “Post-Quantum Cryptography for Quantum Resilient Data Transfer”. In: *2025 25th Anniversary International Conference on Transparent Optical Networks (ICTON)*. ed. by C. Cojocar, S. Spadaro, and M. Marciniak. United States: Institute of Electrical and Electronics Engineers, Aug. 2025. DOI: 10.1109/ICTON67126.2025.11125195
 - D. Lawo, R. Abu Bakar, A. Cano Aguilera, F. Cugini, J. L. Imaña, I. Tafur Monroy, and J. J. Vegas Olmos. “Future Data Centers: Hardware-Offloaded Post-Quantum Secure Optical IPsec Tunnel for Confidential AI Training”. In: *Proceedings of the 2025 International Conference on Optical Network Design and Modeling (ONDM)*. Pisa, Italy, 2025. URL: <https://opendl.ifip-tc6.org/db/conf/ondm2025/ondm2025/1571114929.pdf>
 - D. Iliadis-Apostolidis, D. C. Lawo, S. Kosta, I. Tafur Monroy, and J. J. Vegas Olmos. “QRoNS: Quantum Resilience Over IPsec Tunnels for Network Slicing”. In: *Electronics* 14.21 (2025), p. 4234. DOI: 10.3390/electronics14214234
 - D. Iliadis-Apostolidis, D. C. Lawo, S. Kosta, and J. J. Vegas Olmos. “SNSVS: Scalable Network Slicing with Virtualized Systems using Post-Quantum Cryptography”. In: *GLOBE-COM 2025 - 2025 IEEE Global Communications Conference*. IEEE, 2025, pp. 3109–3114. DOI: 10.1109/GLOBECOM59602.2025.11431714
 - D. C. Lawo, I. T. Monroy, and J. J. V. Olmos. “Confidential Computing and AI in Networking Fabrics”. In: *2025 25th Anniversary International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2025, pp. 1–4. DOI: 10.1109/ICTON67126.2025.11125026
 - J. Meyer, A. Cano, F. Cugini, I. Tafur Monroy, and J. J. Vegas Olmos. *Design and Evaluation of a Heterogeneous DPU Architecture for Accelerating Post-Quantum Cryptography*. Preprint, version 1, Research Square. Nov. 2025. DOI: 10.21203/rs.3.rs-7817696/v1

Material derived from these theses and publications has been reorganized, synthesized, updated, and integrated into the broader technical narrative of the book. The book also draws on standards, protocol specifications, technical reports, foundational research, and independent studies cited throughout the individual chapters; these sources are listed in the bibliography.

Figures and tables adapted or reproduced from previously published sources are identified in their captions together with the relevant citation and, where applicable, licensing or permission information.

Acknowledgements

The authors gratefully acknowledge Athanasios Panagiotis Pelekoudas for manuscript preparation, technical writing, synthesis, and LaTeX development, and Emilia Castillo Frederiksen for the preparation of figures, tables, flowcharts, and other visual material, as well as copyright and permissions coordination.

Contents

Source and Attribution	ii
1 Introduction	1
1.1 The Emerging Challenge of Quantum Computing	1
1.2 Why Cryptography Is an Infrastructure Problem	3
1.3 From Classical to Post-Quantum Cryptography	4
1.4 From Theory to Practice	6
1.5 Scope and Contributions of This Book	7
1.6 Assumed Background	8
1.7 Organization of the Book	9
2 Classical Cryptography and System Assumptions	10
2.1 Security Foundations and Computational Assumptions	10
2.2 Symmetric Cryptography and Data Protection	12
2.3 Public-Key Cryptography and Trust Establishment	17
2.4 Cryptography in Communication Protocols and Systems	22
2.5 Stability of Classical Cryptographic Assumptions	26
3 The Quantum Threat Model	29
3.1 Quantum Computing and the Security Model	29
3.2 Quantum Attacks on Classical Cryptography	30
3.3 What Quantum Adversaries Change in Systems	34
3.4 The Urgency of the Quantum Threat	35
3.5 Why Classical Cryptography Cannot Be Incrementally Fixed	37

4	Post-Quantum Cryptography in Transition	39
4.1	Post-Quantum Cryptography and New Foundations	39
4.2	Trade-offs in Post-Quantum Cryptography	45
4.3	Hybrid Cryptography and the Transition to Post-Quantum Systems	52
4.4	Migrating to Post-Quantum Cryptography	56
4.5	What Post-Quantum Migration Means for Systems	61
5	From Infrastructure to Application in Post-Quantum Cryptography	63
5.1	Cryptography in Modern Infrastructure	63
5.2	Infrastructure as a Constraint on Post-Quantum Cryptography	64
5.3	Infrastructure-Level and Applied Post-Quantum Cryptography	67
5.4	What Applied Post-Quantum Cryptography Means in Practice	68
6	Applied Post-Quantum Cryptography in Practice	72
6.1	From Cryptographic Algorithms to System Deployment	72
6.2	Post-Quantum Cryptography in the Networking Stack	74
6.3	Performance and Scalability in Post-Quantum Systems	81
6.4	Hardware Acceleration and Cryptographic Offloading	92
6.5	Post-Quantum Cryptography in Virtualized and Multi-Tenant Systems	99
7	Discussion and Practical Implications	105
7.1	From Algorithmic Security to System-Level Deployment	105
7.2	Lessons from Infrastructure-Level and Applied Post-Quantum Cryptography	107
7.3	Practical Implications for Secure Computing Infrastructures	109
7.4	Future Directions for Quantum-Resilient Architectures	112
7.5	Broader Post-Quantum Migration Landscape	114
7.6	Concluding Remarks	116

Chapter 1

Introduction

1.1 The Emerging Challenge of Quantum Computing

Modern digital infrastructure is defined by continuous, high-volume data exchange across global communication systems, cloud platforms, data centers, and distributed applications. As the volume, sensitivity, and speed of this exchange continue to increase, cryptography becomes indispensable for preserving confidentiality, integrity, authentication, and trust. These mechanisms are deeply embedded in protocols, software stacks, hardware platforms, and large-scale trust infrastructures, forming a critical foundation for secure computation and communication.

Yet the security of widely deployed cryptographic systems rests on computational assumptions that are increasingly challenged by the prospective capabilities of quantum computing. Algorithms such as Shor’s algorithm show that problems underlying classical public-key cryptography, including integer factorization and discrete logarithms, can be solved efficiently on a sufficiently powerful quantum computer [22]. Consequently, mechanisms that currently secure the Internet, cloud infrastructures, and distributed systems may no longer provide long-term protection, making the transition toward quantum-resistant alternatives both necessary and infrastructurally disruptive.

This challenge emerges at a time when modern computing workloads are also becoming increasingly data-intensive. Applications such as Artificial Intelligence (AI), large-scale analytics, and distributed high-performance computing require continuous data movement between storage, compute nodes, accelerators, and network devices. As network technologies advance toward hundreds of gigabits and terabit-scale transmission rates, general-purpose processing capabilities have not scaled at the same pace. This growing mismatch between data movement and data processing makes cryptographic overhead a system-level concern, especially when new cryptographic mechanisms introduce larger artifacts and more demanding computations [23, 24, 25].

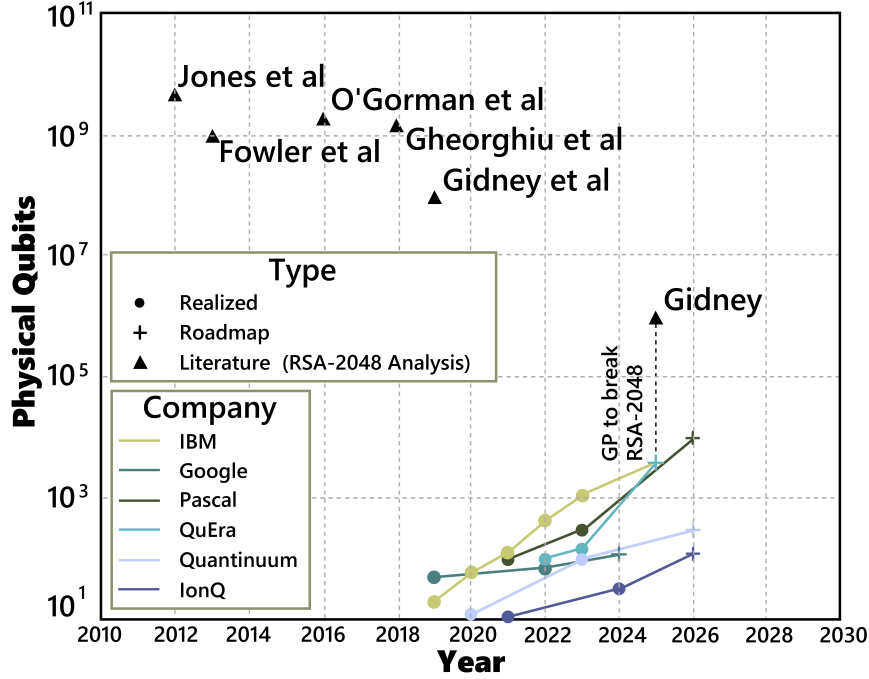


Figure 1.1: Vendor roadmaps for physical qubit counts compared with selected resource estimates for factoring Rivest–Shamir–Adleman (RSA)-2048. The comparison is illustrative rather than predictive, since physical-qubit roadmaps do not directly determine the availability of fault-tolerant logical qubits, and factoring estimates depend on error-correction, runtime, and architectural assumptions. Adapted from Figure 1.3 of [1], with permission from the author. Data from [26, 27].

The projected evolution of quantum computing capabilities further illustrates why the transition to quantum-resistant cryptography is treated as a long-term infrastructure-planning problem. As shown in Figure 1.1, vendor roadmaps indicate a steady increase in physical qubit counts across different quantum computing platforms. These roadmaps can be compared with resource estimates for factoring RSA-2048, but such comparisons must be interpreted carefully. Physical qubit counts do not directly correspond to the logical, fault-tolerant qubits required to execute Shor’s algorithm at cryptographically relevant scale, since the overhead of error correction, gate fidelity, runtime assumptions, and architecture design remains decisive. Earlier estimates by *Gidney* and *Ekerå* suggested that RSA-2048 could be factored in eight hours using approximately 20 million noisy physical qubits. *Gidney*’s 2025 revision reduced the estimated requirement to fewer than one million noisy physical qubits, with a runtime of less than one week, while retaining the same baseline physical-hardware assumptions used in the earlier estimate. The reduction instead reflects advances in the factoring construction and fault-tolerant implementation, together with a different space–time trade-off [26, 28]. These revisions show that the relevant resource threshold is not fixed, but can change substantially as algorithms and fault-tolerant implementation techniques improve.

The urgency of this transition is also shaped by Harvest Now, Decrypt Later (HNDL), since data that must remain confidential for many years may require quantum-resilient protection before large-scale quantum computers exist [29]. Chapter 3 develops this time-sensitive threat model in detail.

1.2 Why Cryptography Is an Infrastructure Problem

Cryptography is often introduced as a collection of mathematical primitives that can be designed, analyzed, and evaluated independently. In operational settings, however, cryptographic mechanisms are embedded within the communication stack, operating system networking paths, hardware acceleration engines, and large-scale trust frameworks such as Public Key Infrastructure (PKI). They appear at multiple layers, from link-layer protection to transport and application-level protocols. As a result, cryptography is not an isolated component, but a distributed capability that operates continuously as part of the data path.

In this context, infrastructure refers not only to physical networks and computing hardware, but also to the protocol stacks, operating-system mechanisms, trust frameworks, virtualization layers, cryptographic libraries, and operational workflows that allow secure communication to function at scale. A cryptographic mechanism therefore becomes infrastructural when its assumptions are embedded into these layers and when changing it requires coordinated adaptation across them.

This embeddedness has important consequences. Changes to cryptographic primitives do not remain confined to individual algorithms. Instead, they propagate across protocols, software implementations, and hardware execution paths. For instance, authentication and key-establishment mechanisms are encoded into widely deployed protocols such as Transport Layer Security (TLS), Internet Protocol Security (IPsec), and Media Access Control Security (MACsec). These protocols define message formats, processing steps, and communication flows that depend on specific cryptographic properties, including key sizes and computational cost. Modifying these assumptions requires coordinated changes across the system, rather than simple algorithm replacement. The same is true for trust infrastructures such as PKI, where certificate formats, signature algorithms, certificate-chain validation, and trust-anchor management are directly affected by changes in public-key cryptography.

Historical experience further illustrates the complexity of cryptographic transitions. Legacy mechanisms can remain in deployed systems long after stronger replacements become available because compatibility requirements, software dependencies, procurement cycles, and coordinated upgrade costs slow migration [30, 31]. This inertia makes cryptographic replacement an operational and infrastructural problem as well as a technical one.

For decades, cryptographic evolution followed a model of gradual adaptation. As computational power increased, security could be maintained by adjusting parameters, such as increasing key sizes, without changing system architecture. This approach depended on the stability of the underlying computational model. Quantum computing changes this assumption. Rather than merely weakening cryptographic schemes, quantum algorithms introduce a structural discontinuity by invalidating the hardness assumptions behind widely deployed public-key systems. As a result, entire classes of cryptographic primitives must be replaced, rather than adjusted.

In this context, Post-Quantum Cryptography (PQC) is not simply an algorithmic upgrade, but a system-wide transformation. Post-quantum schemes introduce new computational profiles and substantially larger cryptographic artifacts, affecting protocol message formats, software implementations, hardware execution paths, communication efficiency, and scalability. In large-scale environments, where cryptographic operations are executed across millions of connections, even small changes can have significant system-wide effects. Their deployment is therefore shaped by protocol architectures, processing capabilities, and operational environments. Consequently, the transition to PQC requires coordinated adaptation across algorithms, protocols, software stacks, and hardware platforms.

This also makes crypto-agility a central requirement. Systems must be able to negotiate, replace, combine, and deprecate cryptographic mechanisms without requiring complete architectural redesign. In the post-quantum transition, this flexibility is necessary because standards, implementations, and cryptanalytic confidence will continue to evolve [32, 33].

1.3 From Classical to Post-Quantum Cryptography

Classical cryptography has provided the foundation for secure digital communication for several decades. Widely deployed mechanisms such as RSA and Elliptic-Curve Cryptography (ECC) rely on mathematical problems that are considered infeasible to solve with classical computers. These include integer factorization and the discrete logarithm problem, which underpin most public-key systems used for authentication, key exchange, and digital signatures. In deployed systems, these mechanisms are combined with symmetric encryption to establish trust efficiently while protecting bulk data at high throughput.

The security of these systems depends on the assumption that certain mathematical problems remain infeasible to solve within realistic computational limits. This assumption has guided the design of cryptographic protocols across the Internet and modern computing infrastructures. However, the emergence of quantum computing challenges this foundation. Quantum algorithms, most notably Shor's algorithm, can solve these problems in polynomial time, making classical public-key mechanisms insecure once large-scale quantum computers become available [22].

The quantum threat does not affect all cryptographic mechanisms equally. Symmetric primitives and hash functions are mainly affected through reduced effective security levels, which can often be mitigated with larger parameters. By contrast, public-key mechanisms based on integer factorization and discrete logarithms are structurally vulnerable, since their underlying hardness assumptions can be invalidated by quantum algorithms. This distinction explains why post-quantum migration primarily concerns authentication, key establishment, digital signatures, certificates, and trust infrastructures [22, 34].

This development has led to the field of PQC, which focuses on designing mechanisms that remain secure against both classical and quantum adversaries. Instead of relying on factorization or discrete logarithms, post-quantum schemes draw on alternative problem families for which no efficient quantum attacks are currently known. These include lattice-based constructions, hash-based signatures, and code-based cryptographic systems [35]. Each approach introduces different trade-offs in terms of computational cost, key size, and security assumptions.

The transition is also shaped by ongoing standardization. The first National Institute of Standards and Technology (NIST) post-quantum standards define Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM), based on the CRYSTALS-Kyber submission, Module-Lattice-Based Digital Signature Algorithm (ML-DSA), based on CRYSTALS-Dilithium, and Stateless Hash-Based Digital Signature Algorithm (SLH-DSA), based on SPHINCS+ [36, 37, 38]. These standardized names are used throughout the book when referring to the finalized NIST algorithms. These mechanisms provide practical building blocks for migration, but they do not remove the need for system-level evaluation. Deployment choices remain dependent on artifact size, computational cost, implementation complexity, and cryptanalytic maturity.

In this book, PQC refers to cryptographic mechanisms designed to run on classical computing infrastructure while remaining secure against quantum adversaries. This distinguishes PQC from Quantum Key Distribution (QKD), which uses quantum communication channels to distribute secret keys. Although QKD may complement post-quantum mechanisms in specialized deployments, it does not replace PQC, since it does not provide general-purpose authentication, digital signatures, or scalable integration across existing digital infrastructures [39].

The adoption of post-quantum schemes, however, introduces a fundamentally different design space for secure system construction. Post-quantum schemes often require more demanding computations and larger cryptographic artifacts, including public keys, signatures, and ciphertexts. Accordingly, their deployment affects not only the execution of cryptographic operations, but also their integration into communication protocols and system architectures [9].

For this reason, the transition to PQC is not simply a replacement of one set of algorithms with another. It reflects a broader shift in how cryptographic mechanisms behave within

real systems. As a result, understanding PQC requires considering both its mathematical foundations and its interaction with practical deployment environments. This perspective forms the basis for the system-level analysis developed in the subsequent chapters.

1.4 From Theory to Practice

Although PQC provides the theoretical basis for quantum-resistant cryptographic mechanisms, deployment in operational systems introduces a distinct set of constraints. In theoretical settings, cryptographic schemes are usually evaluated based on their mathematical properties, such as security guarantees and resistance to known attacks. Such analyses necessarily abstract away many of the constraints imposed by real-world systems.

In practice, cryptographic mechanisms operate within communication protocols, software stacks, and hardware platforms. In these settings, computational cost, memory pressure, and communication overhead become decisive. Post-quantum schemes differ from classical approaches in all of these aspects, since they often require more complex computations and larger cryptographic artifacts.

These constraints become particularly visible once post-quantum mechanisms are incorporated into widely deployed protocols. Protocols such as TLS, IPsec, and MACsec rely on cryptography for authentication, key establishment, and data protection. Incorporating post-quantum mechanisms modifies the establishment of secure sessions and, indirectly, the transmission of protected data. Larger messages increase bandwidth usage and may lead to fragmentation, while more complex computations can introduce additional latency during connection setup [9, 40].

At the same time, modern computing environments operate at very high data rates, especially in cloud and data center infrastructures. In these environments, cryptographic processing occurs continuously and at scale. Even small increases in computational cost or data size can accumulate across many connections, affecting overall system throughput and resource usage. For this reason, performance becomes a key factor in determining whether a post-quantum scheme can be deployed in practice.

These constraints also have architectural consequences. Secure protocols separate relatively infrequent session-establishment functions from continuous data protection, and post-quantum migration changes the cost and implementation of this separation.

In high-throughput and large-scale environments, hardware support becomes central to practical deployment. Specialized components such as Data Processing Units (DPUs) and Smart Network Interface Cards (SmartNICs) can offload selected cryptographic and packet-processing tasks from general-purpose Central Processing Units (CPUs). This allows systems to sustain high data rates while managing the additional complexity introduced by post-quantum mechanisms. As a result, cryptography becomes closely tied to system

architecture, and its efficient deployment depends on the interaction between algorithms, protocols, and hardware platforms [7, 21].

These observations show that the movement from theoretical security to practical deployment is non-trivial. PQC must be evaluated not only by its security guarantees, but also by its behavior within real systems, including its impact on performance, scalability, and integration into existing infrastructures. These aspects are essential for designing quantum-resilient systems that remain secure and efficient under operational constraints.

1.5 Scope and Contributions of This Book

This book examines PQC from a system-level perspective. It focuses on how quantum-resistant mechanisms are integrated into modern computing and communication infrastructures. Rather than concentrating exclusively on the mathematical design of cryptographic primitives, it examines how quantum-resistant mechanisms behave across communication protocols, software stacks, hardware platforms, and large-scale distributed environments.

The book is not intended as a comprehensive survey of post-quantum algorithms or as a formal cryptographic monograph. Instead, it provides a technical, research-oriented dissemination of post-quantum cryptography as a deployment and infrastructure problem. Mathematical foundations are introduced where needed to explain why new constructions are required, but the main focus is on how these constructions interact with real systems.

The scope of the book is centered on applied and infrastructure-level aspects of PQC. It considers how cryptographic mechanisms interact with protocol architectures such as TLS, IPsec, and MACsec. It also examines how they affect performance, communication efficiency, and scalability. In addition, the book discusses the role of hardware acceleration and the deployment of PQC in virtualized and multi-tenant environments, where isolation and efficient resource use are essential.

For clarity, the book distinguishes between three levels of analysis. Theoretical PQC concerns the mathematical construction and formal security of quantum-resistant primitives. Infrastructure-level PQC concerns the integration of those primitives into protocols, trust frameworks, key-management systems, and hardware-supported architectures. Applied PQC concerns their practical execution and observed behavior under real deployment constraints, including latency, throughput, memory pressure, scalability, interoperability, and isolation.

The central contribution of this book is the explicit framing of PQC as an infrastructure-level problem. From this viewpoint, cryptography is not treated as an isolated component, but as part of the infrastructure itself. Feasibility therefore depends not only on formal security, but also on how efficiently cryptographic mechanisms operate within real sys-

tems. The interaction between algorithms, protocols, hardware platforms, and deployment environments is consequently examined as a single connected problem.

More specifically, the book contributes four connected perspectives. First, it explains why the quantum threat changes the computational assumptions underlying classical public-key infrastructure. Second, it presents post-quantum migration as a transition involving new mathematical foundations, hybrid mechanisms, and crypto-agile design. Third, it analyzes how infrastructure constrains deployment through protocol formats, packet-processing paths, hardware acceleration, and trust-management systems. Fourth, it examines applied deployments in networking, virtualized, and multi-tenant environments to examine how quantum-resilient security can be deployed while managing performance and scalability constraints.

Overall, the book connects theoretical cryptographic research with practical system deployment. By examining PQC across multiple system layers, it shows how quantum-resilient security can be realized in modern infrastructures. This perspective is intended to support researchers and practitioners in understanding the main challenges and design choices involved in deploying PQC at scale.

1.6 Assumed Background

This book is intended for readers with a technical background in cybersecurity, computer science, networking, or related fields. Basic familiarity with discrete mathematics, linear algebra, probability, and layered communication systems is helpful, but prior formal study of post-quantum cryptography is not required.

Chapter 2 introduces the classical cryptographic concepts, security assumptions, and notation needed for the subsequent discussion. Chapter 4 introduces the additional mathematical foundations required for lattice-based post-quantum cryptography, including lattices, Learning With Errors (LWE), Module Learning With Errors (MLWE), and Short Integer Solution (SIS). The later chapters then shift toward protocols, infrastructure, hardware, and applied deployment.

The mathematical material is included to establish the foundations required for the system-level analysis rather than to provide a complete formal treatment of modern cryptography. Readers primarily interested in infrastructure and deployment may therefore use the mathematical definitions in Chapters 2 and 4 as supporting reference material for the applied discussion that follows.

1.7 Organization of the Book

The book is structured to guide the reader from the foundations of classical cryptography to the practical implications of deploying post-quantum mechanisms in modern infrastructures. It first establishes the assumptions on which contemporary secure systems rely, then examines how quantum computing challenges these assumptions, and finally analyzes how post-quantum cryptography can be integrated into real systems.

Chapter 2 establishes the foundations of classical cryptography and the system assumptions that have shaped modern digital infrastructure. It discusses computational security, symmetric and public-key cryptography, trust establishment, and the use of cryptography in communication protocols such as TLS, IPsec, and MACsec.

Chapter 3 analyzes the quantum threat model. It explains how quantum computing changes the security assumptions of classical cryptography, analyzes the effects of quantum algorithms on symmetric and asymmetric mechanisms, and discusses why classical cryptographic systems cannot be incrementally fixed.

Chapter 4 presents post-quantum cryptography as a transitional process. It introduces the main quantum-resistant foundations, discusses the trade-offs associated with post-quantum schemes, and examines hybrid cryptography, migration strategies, and the system-level consequences of moving toward quantum-resilient infrastructures.

Chapter 5 shifts the focus from cryptographic mechanisms to infrastructure. It analyzes how cryptography operates within modern systems, explains why infrastructure constrains post-quantum deployment, and introduces the distinction between infrastructure-level and applied post-quantum cryptography.

Chapter 6 develops applied post-quantum cryptography in practice. It considers how quantum-resistant mechanisms are integrated into the networking stack, how they affect performance and scalability, and how hardware acceleration, cryptographic offloading, virtualization, and multi-tenant environments influence deployment.

Chapter 7 discusses the broader implications of the preceding analysis. It synthesizes the main lessons from infrastructure-level and applied post-quantum cryptography, identifies practical implications for secure computing infrastructures, and outlines future directions for quantum-resilient architectures.

Across these chapters, post-quantum cryptography is presented not only as a set of mathematical constructions, but as a system-level capability that must be integrated, managed, and updated across protocols, software stacks, hardware platforms, trust infrastructures, and operational environments.

Chapter 2

Classical Cryptography and System Assumptions

This chapter examines the foundations of classical cryptography and the assumptions that have shaped modern digital infrastructures. It first introduces the computational security model and the distinction between symmetric and public-key mechanisms. It then explains how these mechanisms are combined in communication protocols such as TLS, IPsec, and MACsec. The chapter concludes by examining why classical assumptions remained stable for decades and why that stability is now being reconsidered in light of emerging computational models.

Source note. Parts of this chapter draw substantially on Chapter 2 of [1]. Selected protocol and networking material also draws on Chapters 5–6 of [1] and Chapter 3 of [2]. The material has been reorganized and supplemented with the standards and other sources cited throughout.

2.1 Security Foundations and Computational Assumptions

Modern cryptography defines security as a formally verifiable property rather than as obscurity or concealment. A cryptographic system is considered secure when legitimate users can operate it efficiently, while unauthorized parties cannot compromise it within feasible computational limits [41]. Within a precisely defined mathematical model, this means that no feasible adversary can recover protected information or violate the intended guarantees. Security is therefore grounded in explicit assumptions about attacker capabilities and in formally defined guarantees that hold under those assumptions. This shift from informal protection mechanisms to mathematically grounded models marks the foundation of modern digital cryptography.

Security in modern cryptography is understood through two principal notions, namely unconditional security and computational security. Unconditional or perfect security represents the strongest theoretical notion of protection in cryptography, describing schemes that remain secure even against adversaries with unlimited computational power [41, 42].

Let E be a cipher defined over key space $\mathcal{K}$, message space $\mathcal{M}$, and ciphertext space $\mathcal{C}$. The cipher achieves perfect secrecy if the ciphertext reveals no information about the underlying plaintext. Formally, for all $m_0, m_1 \in \mathcal{M}$ and all $c \in \mathcal{C}$,

$$\Pr[E(k, m_0) = c] = \Pr[E(k, m_1) = c],$$

where the probability is taken over uniformly distributed keys $k \in \mathcal{K}$ [41].

In other words, observing the ciphertext does not change the adversary's probability distribution over possible messages. An example of an unconditionally secure cipher is the One-Time Pad [43], where encryption and decryption are defined as

$$E(m, k) = m \oplus k, \quad D(c, k) = c \oplus k,$$

using a uniformly random key k that is as long as the message and never reused [41]. Under these conditions, the ciphertext is statistically independent of the plaintext.

Despite its theoretical elegance, unconditional security is not suitable for large-scale digital systems. Perfect secrecy requires secret keys that are at least as long as the data being protected, and those keys must be distributed securely and never reused. In contemporary communication infrastructures, where massive volumes of data are exchanged continuously across distributed systems, these requirements impose prohibitive constraints on key generation, distribution, and storage.

For these reasons, modern cryptography is grounded in the computational security paradigm. Unlike perfect secrecy, computational security does not require ciphertext distributions to be statistically identical. Instead, it requires that no efficient adversary can exploit their differences with more than negligible advantage. Security is therefore defined relative to efficient adversaries and a security parameter λ .

For an encryption scheme with encryption algorithm Enc , this idea can be expressed through message indistinguishability. Let $m_0, m_1 \in \mathcal{M}$ be two equal-length messages. A key k is generated according to the scheme, a bit $b \leftarrow \{0, 1\}$ is chosen uniformly at random, and the adversary is given a ciphertext

$$c \leftarrow \text{Enc}_k(m_b).$$

The adversary $\mathcal{A}$ outputs a guess b' for the hidden bit b . Its distinguishing advantage is

$$\text{Adv}_{\mathcal{A}}(\lambda) = \left| \Pr[b' = b] - \frac{1}{2} \right|.$$

A computationally secure encryption scheme requires this advantage to be negligible in λ for every probabilistic polynomial-time adversary $\mathcal{A}$ [41].

This asymptotic notion should be distinguished from the concrete notion of an n -bit security level. Informally, saying that a construction provides 128-bit security means that the best known attack is expected to require on the order of 2^{128} elementary operations under the stated attack model. Increasing the security parameter therefore increases the computational resources required to compromise the scheme [42]. Crucially, this model decouples the length of protected data from the size of secret keys, allowing relatively short keys to secure arbitrarily large volumes of information. This property made large-scale secure communication feasible and enabled the deployment of cryptographic primitives such as Advanced Encryption Standard (AES), RSA, and elliptic-curve constructions in global infrastructures.

This model made large-scale cryptographic deployment practical. By allowing security to be adjusted through parameters such as key size, computational security enabled cryptographic primitives to be standardized, implemented, and embedded into communication protocols and digital infrastructures. The long-term stability of these assumptions is examined in Section 2.5.

2.2 Symmetric Cryptography and Data Protection

Symmetric cryptography forms the computational core of data protection in modern digital systems. In its standard formalization, a symmetric encryption scheme is defined as a tuple of probabilistic polynomial-time algorithms

$$\mathcal{E} = (\text{KeyGen}, \text{Enc}, \text{Dec}),$$

where $\text{KeyGen}(1^\lambda)$ generates a secret key k drawn from a key space determined by the security parameter λ , $\text{Enc}_k(m)$ produces a ciphertext c from a plaintext message m , and $\text{Dec}_k(c)$ recovers the original message such that

$$\text{Dec}_k(\text{Enc}_k(m)) = m$$

for all valid messages m [41]. The defining characteristic of symmetric cryptography is that the same shared key k is used for both protection and recovery of data. Once such a key is established between two parties, large volumes of information can be processed efficiently under this shared secret.

A dominant symmetric primitive in deployed systems is the block cipher. Such ciphers implement keyed permutations over fixed-size input blocks, typically 128 bits in the case of AES [44]. AES follows a substitution–permutation structure composed of repeated keyed transformations, applied over multiple rounds during the encryption process. These transformations are designed to achieve diffusion and nonlinearity across the block [45]. Standardized by NIST, AES supports key sizes of 128, 192, and 256 bits, allowing security to scale through parameter selection without altering the underlying construction [44]. Its widespread adoption stems from strong security assurances, computational efficiency, and ubiquitous support across software libraries and hardware platforms. Figure 2.1 illustrates the internal structure of AES, showing how plaintext is transformed into ciphertext through a sequence of round-based operations.

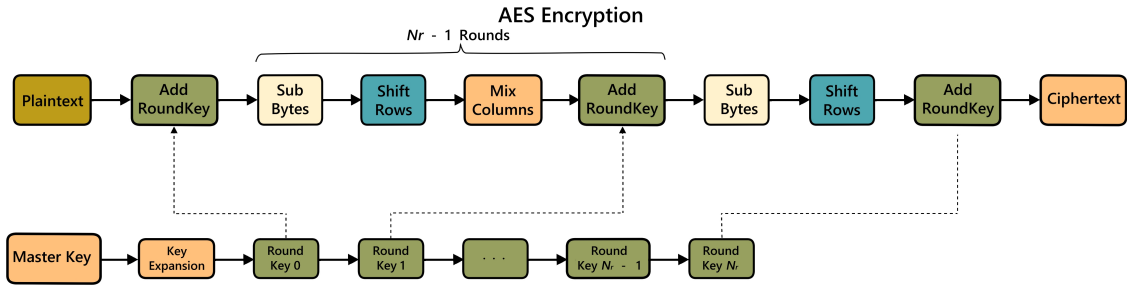


Figure 2.1: Structure of AES encryption process. Adapted from Figure 2.1 of [1], with permission from the author.

The encryption process consists of multiple rounds, each applying a combination of substitution, permutation, and key-mixing operations. These include nonlinear substitution (*SubBytes*), permutation across rows (*ShiftRows*), linear mixing of columns (*MixColumns*), and the addition of round keys derived from the master key. Together, these transformations ensure that the ciphertext depends on both the key and the input data in a highly nontrivial way. Having established how a single block is processed, it is important to consider how block ciphers are applied to longer messages.

A block cipher, however, operates only on fixed-size inputs. To encrypt messages of arbitrary length, a mode of operation is required. Standardized block cipher modes specify how successive blocks are processed and combined to produce secure ciphertext streams [46]. Early constructions such as Cipher Block Chaining (CBC) provided confidentiality but required separate mechanisms for integrity. Modern high-performance systems increasingly rely on counter-based constructions, particularly Galois/Counter Mode (GCM), which transforms a block cipher into a parallelizable stream-like construction while simultaneously generating authentication tags [47].

In addition to encryption primitives, modern symmetric cryptography relies heavily on cryptographic hash functions. A hash function is a deterministic and efficiently computable function H that maps an input of arbitrary length to a fixed-length output, commonly

referred to as a digest [41].

$$H : \{0, 1\}^* \rightarrow \{0, 1\}^n$$

A cryptographic hash function must satisfy three main security properties. Preimage resistance means that, given a digest y , it is computationally infeasible to find an input x such that $H(x) = y$. Second preimage resistance means that, given an input x , it is difficult to find a different input $\hat{x}$ such that $H(x) = H(\hat{x})$. Collision resistance means that it is hard to find any two distinct inputs $x, \hat{x}$ such that $H(x) = H(\hat{x})$. These properties make hash functions a fundamental building block for ensuring data integrity, authentication, and efficient verification in modern cryptographic systems [41].

Confidentiality alone is insufficient in adversarial environments where attackers may modify or inject messages. Integrity and origin authentication can be provided through Message Authentication Codes (MACs), which use a shared secret key to compute a short authentication tag over a message. Replay protection is a separate protocol property and normally requires freshness information, such as a nonce, counter, or sequence number, to be included in the authenticated data [1, 41].

Formally, a MAC scheme consists of a tag-generation algorithm **Tag** and a verification algorithm **Verify**. Given a secret key k and a message m , the sender computes

$$\tau \leftarrow \text{Tag}_k(m),$$

and the receiver accepts only if

$$\text{Verify}_k(m, \tau) = 1.$$

Correctness requires honestly generated tags to verify, while security requires that an adversary without the secret key cannot feasibly produce a valid tag for a new message. Unlike a digital signature, a MAC is symmetric: tag generation and verification depend on the same shared key, so it does not provide public verifiability or non-repudiation [41].

Common constructions include Hash-based Message Authentication Code (HMAC), which derives authentication from a cryptographic hash function, and Cipher-based Message Authentication Code (CMAC), which constructs authentication from a block cipher such as AES [1, 48, 49].

Authenticated Encryption with Associated Data (AEAD) integrates confidentiality and integrity into a single primitive [41]. An AEAD scheme encrypts a message while simultaneously authenticating both the encrypted content and selected metadata, referred to as associated data. This design is particularly important in network protocols, where header fields must remain visible for routing but must still be protected against tampering [47].

AES-GCM is widely deployed in protocols such as TLS, IPsec, and MACsec, providing authenticated encryption in a construction that can be implemented efficiently [50, 51, 52].

Figure 2.2 illustrates how AES-GCM combines encryption and authentication within a single construction.

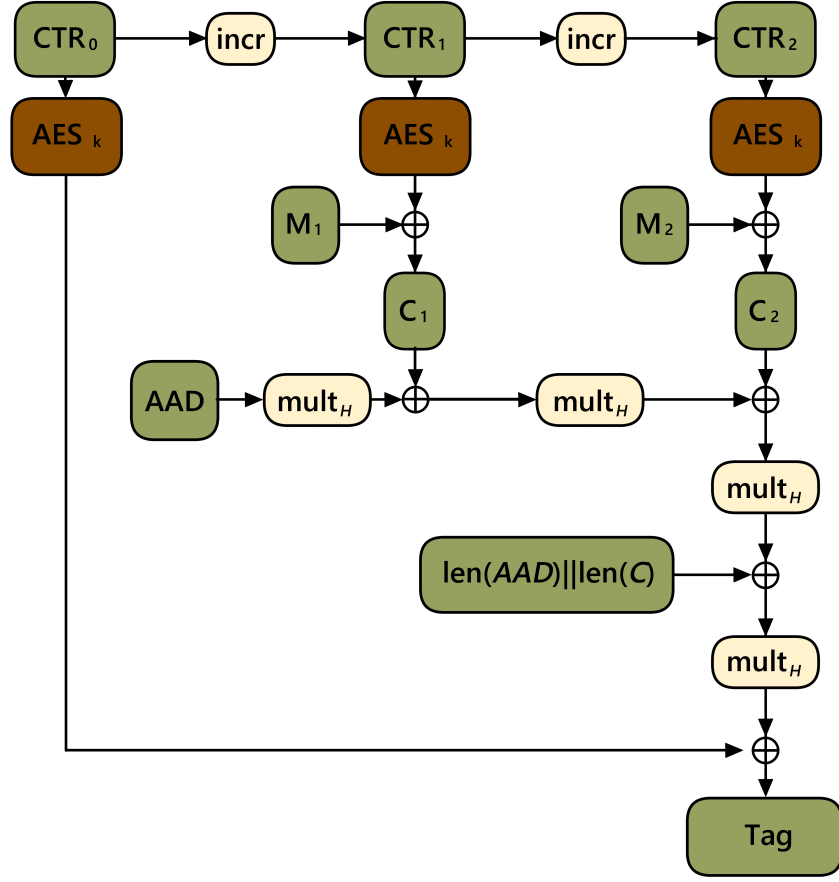


Figure 2.2: Operation of AES-GCM. Adapted from Figure 2.5 of [1], with permission from the author.

This construction combines two main processes. Encryption is performed using AES in counter mode, where successive counter values are encrypted and combined with the plaintext to produce ciphertext blocks. In parallel, an authentication function operates over both the associated data and the ciphertext, producing a final authentication tag. This construction ensures that both the encrypted content and the associated metadata are protected against modification [47].

In addition to encryption and authentication, symmetric systems rely on Key Derivation Functions (KDFs) to transform shared secrets into cryptographically strong subkeys. A KDF derives multiple independent keys from an initial secret, such as separate keys for encryption, authentication, or bidirectional communication. This separation between key establishment and key expansion supports modular protocol design [53].

While traditional hash functions produce fixed-length outputs, many modern cryptographic constructions require larger amounts of pseudorandom material. This is particularly

relevant in post-quantum cryptography, where algorithms often rely on sampling structured objects that exceed the size of standard hash outputs. For this reason, Extendable-Output Functions (XOFs) have been introduced as a generalization of hash functions. An XOF is a deterministic function

$$X : \{0, 1\}^* \rightarrow \{0, 1\}^*$$

that produces an output of arbitrary length selected by the user [54]. Unlike fixed-length hash functions, XOFs allow the output length to be selected according to the requirements of the application. Their achievable security strength depends on the construction and the selected output length, and their use is therefore subject to security considerations that differ from those of conventional fixed-length hash functions [54].

The Secure Hash Algorithm KECCAK (SHAKE) functions defined in the Secure Hash Algorithm 3 (SHA-3) standard are based on sponge constructions [54]. These constructions operate in two phases. During absorption, the input is incorporated into an internal state. During squeezing, output bits are generated from that state. This design enables efficient and secure generation of large amounts of pseudorandom data, making XOFs particularly suitable for post-quantum cryptographic schemes and key derivation processes.

Symmetric cryptography is substantially less expensive than public-key cryptography for bulk data protection and can sustain high throughput in optimized software and hardware implementations [1]. Modern processors provide dedicated instruction-set support for AES, while some Network Interface Cards (NICs) and DPUs provide hardware engines for high-rate encryption [2, 14]. This hardware support allows symmetric data protection to be integrated efficiently into performance-sensitive communication paths.

Beyond raw efficiency, symmetric cryptography scales effectively because its structure allows parallel processing. In counter-based modes such as Counter Mode (CTR) and GCM, each block can be encrypted independently, enabling execution across multi-core CPUs and hardware accelerators. This property allows encryption performance to grow with available computational resources and network bandwidth. As link speeds approach and exceed hundreds of gigabits per second, including emerging 800G and terabit-scale Ethernet standards [25], distributing encryption across parallel processing units helps prevent security processing from becoming a performance bottleneck.

The long-term standardization of AES and related constructions facilitated their deep integration into operating systems, cryptographic libraries, and networking stacks. Because these primitives remained stable and could be strengthened simply by adjusting key lengths, they were embedded into kernel implementations, transport protocols, and hardware accelerators without requiring architectural redesign. Over time, symmetric cryptography became a foundational and largely invisible component of secure communication infrastructure, protecting the data plane while imposing predictable computational costs.

However, symmetric cryptography assumes that a shared secret already exists between the communicating parties. Establishing such secrets across open networks requires a different cryptographic model, which motivates the public-key mechanisms discussed in the next section.

2.3 Public-Key Cryptography and Trust Establishment

Symmetric cryptography efficiently protects data once a shared secret has been established, but it does not solve how such a secret can be created between parties that have never interacted before. In open networks such as the Internet, requiring a pre-shared key between every pair of communicating entities is impractical. The central challenge therefore becomes secure key distribution. The question is how two parties can establish shared secret material over an insecure channel without exposing it to interception.

Public-key cryptography resolves the key distribution challenge by separating encryption capability from decryption capability through the use of key pairs. Formally, a public-key encryption scheme is defined as a triple of probabilistic polynomial-time algorithms ($\text{KeyGen}, \text{Enc}, \text{Dec}$), where $\text{KeyGen}(1^\lambda)$ outputs a key pair (pk, sk) , $\text{Enc}_{pk}(m)$ produces a ciphertext c , and $\text{Dec}_{sk}(c)$ recovers the original message m [41]. A key pair consists of two mathematically related but distinct keys. The public key pk may be distributed openly, whereas the secret key sk must remain confidential. The underlying structure can be understood as a trapdoor function in which computing a ciphertext from a message using the public key is efficient, while recovering the message without the secret key is computationally infeasible for any efficient adversary [41]. Digital signature schemes follow a similar structure, consisting of the algorithms ($\text{KeyGen}, \text{Sign}, \text{Verify}$) that allow a signer to produce a verifiable signature on a message [41]. This asymmetry removes the need to pre-distribute a distinct shared secret between every pair of communicating parties, although authenticating public keys still requires an appropriate trust mechanism.

Many classical public-key encryption constructions can be described using trapdoor one-way functions [41]. A trapdoor function scheme $\mathcal{T}$ consists of three algorithms ($\text{Gen}, \text{Fun}, \text{Inv}$). The algorithm Gen generates a key pair (pk, sk) . The function Fun efficiently computes $y \leftarrow \text{Fun}(pk, x)$ for an input $x \in \mathcal{X}$. The algorithm Inv efficiently recovers $x \leftarrow \text{Inv}(sk, y)$ using the secret key sk . While evaluating $\text{Fun}(pk, x)$ is efficient, recovering x from y without the associated secret key sk remains infeasible for any efficient adversary, with the probability of successful inversion bounded by a negligible function in the security parameter. The secret key sk therefore acts as the trapdoor that enables efficient inversion.

A common mechanism for establishing shared secret material using public-key cryptography is a Key Encapsulation Mechanism (KEM). Formally, a KEM consists of three algorithms ($\text{KeyGen}, \text{Encaps}, \text{Decaps}$). KeyGen generates a key pair (pk, sk) . Given the public key,

$\text{Encaps}(pk)$ produces a ciphertext–shared-secret pair (c, ss) , while $\text{Decaps}(sk, c)$ uses the corresponding secret key to recover the same shared secret ss .

Unlike Diffie–Hellman (DH)-style Key Exchange (KEX), where both parties contribute values from which the shared secret is derived, a KEM follows an asymmetric encapsulation model: one party encapsulates using the recipient’s public key, and the recipient derives the corresponding secret through decapsulation [1, 41]. Figure 2.3 illustrates this process using a trapdoor-based construction.

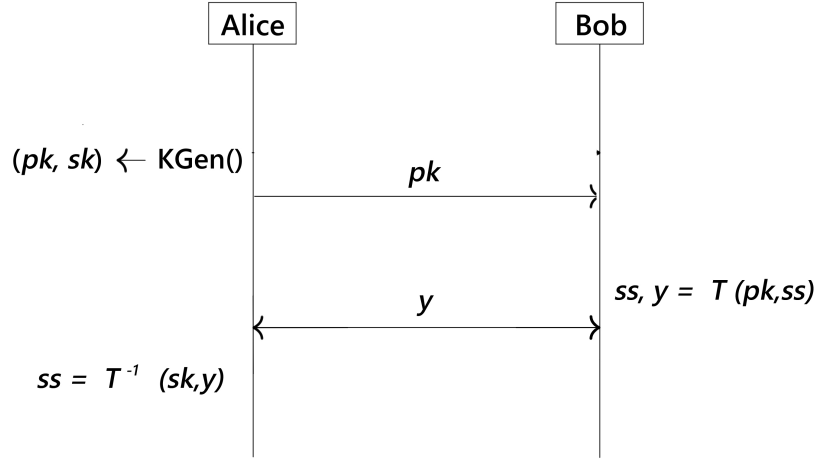


Figure 2.3: Generic key encapsulation mechanism based on a trapdoor function. Adapted from Figure 2.8 of [1], with permission from the author.

Classical public-key constructions instantiate this trapdoor paradigm using number-theoretic hardness assumptions. In RSA-based systems, security relies on the difficulty of the Integer Factorization Problem (IFP). Let p and q be two large secret prime numbers and let $N = p \cdot q$. Recovering the prime factors p and q from N is computationally infeasible for any probabilistic polynomial-time adversary $\mathcal{A}$, meaning that the probability that $\mathcal{A}(N)$ outputs (p, q) is negligible in the security parameter λ [41]. Encryption is typically expressed as

$$c = m^e \bmod N,$$

while decryption requires knowledge of a private exponent derived from the factorization of N .

In practice, RSA has also been used in key establishment mechanisms such as RSA-based key encapsulation. However, these constructions lack an important security property known as Perfect Forward Secrecy (PFS), meaning that if the long-term private key is compromised, previously recorded communications can also be decrypted. Modern protocols instead favor ephemeral DH variants for key establishment, since ephemeral key exchange can provide forward secrecy when used and authenticated appropriately [50].

Similarly, DH-based constructions rely on the hardness of the Discrete Logarithm Problem (DLP). Let $\mathbb{G}$ be a cyclic group of prime order q generated by an element $g \in \mathbb{G}$, and let $\alpha \in \mathbb{Z}_q$ with $u = g^\alpha$. Recovering α from (g, u) is computationally infeasible for any efficient adversary $\mathcal{A}$, meaning that the probability that $\mathcal{A}(g, u)$ outputs α is negligible in the security parameter λ [41]. These hardness assumptions formed the mathematical foundation of classical asymmetric cryptography for decades.

Elliptic-curve-based constructions rely on a related but distinct hardness assumption known as the Elliptic-Curve Discrete Logarithm Problem (ECDLP). Let $E(\mathbb{F}_q)$ be an elliptic curve defined over a finite field, and let P be a point of order n on the curve. Given a point $Q = kP$, recovering the scalar k is computationally infeasible for any efficient adversary [41]. The ECDLP enables comparable classical security with substantially smaller parameters than finite-field and RSA-based public-key systems [42, 55].

Beyond encryption, public-key techniques enable secure key establishment without transmitting secret material directly. The DH protocol allows two parties to establish a shared secret over an insecure channel without revealing that secret to an observer [56]. In the protocol, both parties agree on a cyclic group G with generator g . One party selects a private value α and publishes g^α . The other selects β and publishes g^β . Each party then computes the shared secret $g^{\alpha\beta}$ by exponentiating the received public value with its own private exponent. Although an adversary may observe g^α and g^β , deriving the shared secret $g^{\alpha\beta}$ is assumed to be computationally infeasible for appropriately chosen groups. Solving the underlying discrete logarithm problem would, in particular, allow an adversary to recover the private exponents and break the exchange.

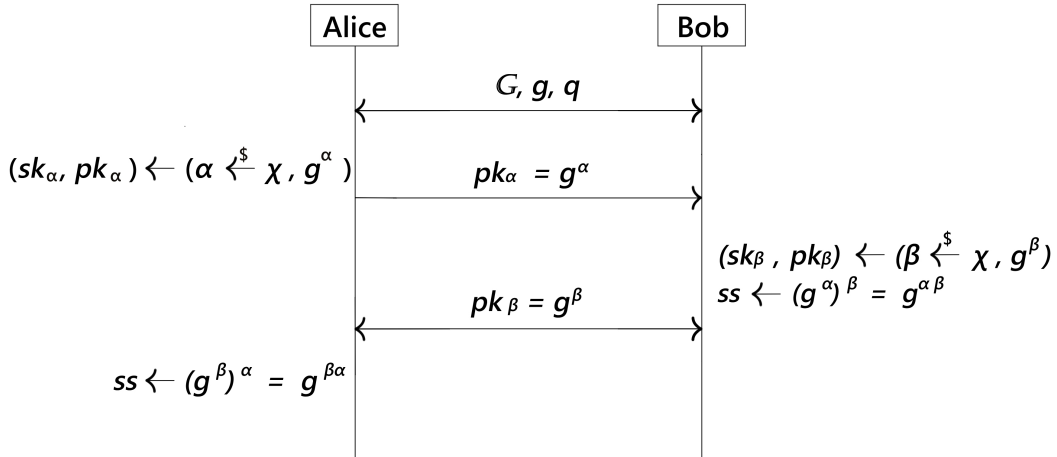


Figure 2.4: DH-based KEX. Adapted from Figure 2.6 of [1], with permission from the author.

Figure 2.4 illustrates the Diffie–Hellman key exchange. Two parties independently generate private values and derive corresponding public values, which are exchanged over an insecure channel. By combining the received public value with their own private value, both parties

compute the same shared secret. Although all exchanged information is publicly visible, recovering the shared secret without solving the underlying DLP remains computationally infeasible.

Elliptic-curve variants (Elliptic-Curve Diffie–Hellman (ECDH)) implement the same structural principle over elliptic-curve groups and replace exponentiation with point multiplication, for example, $ss = \alpha(\beta P)$. This approach achieves equivalent security with substantially smaller parameter sizes and improves efficiency while preserving the same underlying logic [55].

While key exchange establishes shared secrecy, it does not by itself authenticate the communicating parties. Digital signature schemes provide this additional guarantee. Formally, a digital signature scheme is defined as a triple of probabilistic polynomial-time algorithms

$$S = (\text{KeyGen}, \text{Sign}, \text{Verify}),$$

where $\text{KeyGen}(1^\lambda)$ outputs a key pair (pk, sk) for a given security parameter λ . The signing algorithm $\text{Sign}(sk, m)$ produces a signature σ on a message m using the private key sk , and the verification algorithm $\text{Verify}(pk, m, \sigma)$ checks whether σ is a valid signature of m under the public key pk [41]. A signature scheme is secure if it is computationally infeasible for an adversary to generate a valid signature on a new message without knowledge of the signing key.

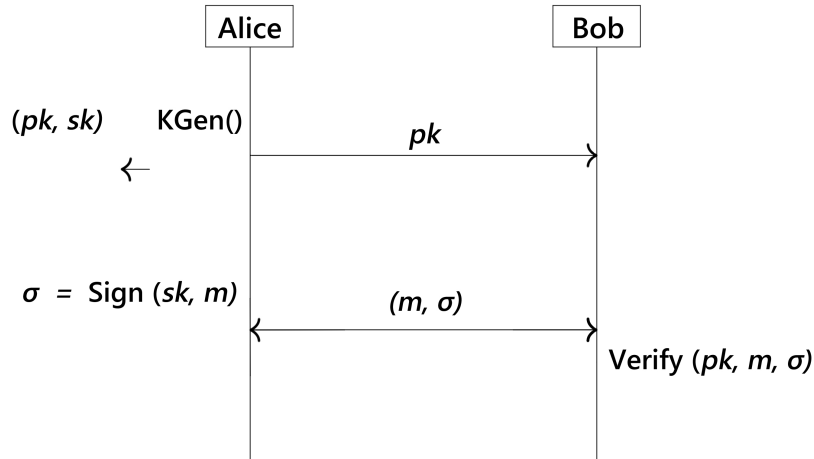


Figure 2.5: Generic digital signature scheme. Adapted from Figure 2.10 of [1], with permission from the author.

Figure 2.5 illustrates the interaction between a signer and a verifier. The signer generates a signature on a message using the private key, while the verifier checks its validity using the corresponding public key. A valid signature shows that the message was signed by the holder of the corresponding private key and that the signed message has not been altered. Binding that key to a real-world identity requires a trust mechanism such as PKI.

Encryption and signatures serve complementary but distinct purposes. Encryption provides confidentiality by allowing only an authorized holder of the decryption key to recover the protected message. Signatures, in contrast, provide integrity and cryptographic evidence that the signer possessed the corresponding private key. When combined with reliable identity binding, secure private-key control, and appropriate operational policy, digital signatures can also support accountability and non-repudiation claims.

Certificate	
Version	e.g. , v3
Serial Number	Unique ID assigned by CA
Signature Algorithm	e.g. , ECDSA with SHA-256
Issuer	Name of the CA
Validity	Not Before / Not After Dates
Subject	Name of Entity (Alice)
Subject Public Key Info - Algorithm (e.g. , ECDSA) - Public Key (Alice's Key)	
Extensions (Optional) - Key Usage - Subject Alt Name (e.g. , DNS, email) - Constraints	
Signature (by CA's private key)	

Figure 2.6: Simplified structure of an X.509 certificate. Adapted from Figure 2.12 of [1], with permission from the author.

At Internet scale, the effectiveness of digital signatures depends on a structured mechanism for binding public keys to identities. PKI provides this framework. PKI defines the roles and procedures required to create, manage, distribute, and revoke digital certificates. Within this framework, a trusted Certificate Authority (CA) verifies an entity's identity and issues a digital certificate that cryptographically binds that identity to a corresponding public key. These certificates follow standardized formats such as X.509 and are digitally signed by the issuing authority, enabling hierarchical trust relationships across distributed systems [57, 58]. Figure 2.6 illustrates the simplified structure of an X.509 certificate. A certificate contains metadata describing both the issuer, typically a CA, and the subject, such as a user or server. A central component is the *Subject Public Key Info* field, which binds the subject's public key to its identity. This binding is secured through the digital signature of the issuing CA, ensuring the authenticity and integrity of the certificate.

Building on this certificate structure, trust is extended through certificate chains. A certificate chain is a hierarchical sequence of certificates that enables an end-entity to be trusted by a party that may have no prior direct relationship with it. An end-entity certificate is digitally signed by an intermediate CA, which in turn is signed by another authority, eventually linking to a root certificate that is self-signed and pre-installed in operating systems and browsers. When a client connects to a server, the server presents its certificate along with the corresponding chain of trust. The client verifies each signature in the chain until it reaches one of the pre-installed trusted root certificates. If each verification step succeeds, the server's identity is authenticated without requiring any prior shared secret or direct trust relationship. This hierarchical validation mechanism enables trust to scale globally while preserving accountability at each level of the chain [57].

For several decades, the hardness assumptions underlying RSA, DH, and related constructions remained computationally stable within the classical model introduced in Section 2.1. As processing capabilities improved, security was preserved by increasing parameter sizes, for example by moving from 1024-bit to 2048-bit RSA keys, without altering the underlying protocol architecture. Asymmetric operations were typically limited to authentication and session establishment, which kept their computational cost manageable, while symmetric encryption protected bulk data. This separation between asymmetric trust establishment and symmetric data protection enabled the development of scalable digital trust across global networks.

2.4 Cryptography in Communication Protocols and Systems

In modern communication systems, cryptographic mechanisms rarely operate in isolation. Instead, they are integrated into structured communication protocols that coordinate authentication, key establishment, and data protection to ensure confidentiality, integrity, and authenticity across open networks [41, 42]. A typical secure communication session begins in an insecure state and gradually transitions into a protected channel through a sequence of cryptographic operations.

In practice, this process often starts with the presentation of a digital certificate by a server, allowing the client to verify its identity through a certificate chain that ultimately leads to a globally trusted root certification authority [57]. Once identity has been established, asymmetric mechanisms such as DH-based KEX are used to negotiate shared secret material over the insecure channel [42, 59]. This exchange allows the communicating parties to derive a common secret without revealing it to external observers.

In practical protocol designs, key establishment is not performed in isolation but as part of an authenticated key exchange. In such constructions, the messages used to derive shared secret material are bound to the identities of the communicating parties, typically through digital signatures or certificate-based authentication. This prevents man-in-the-middle

attacks, in which an adversary intercepts and substitutes key exchange messages in order to establish independent secrets with each party. As a result, the security of the derived session key depends not only on the hardness of the underlying key exchange problem, but also on the correct authentication of the participating entities.

Because asymmetric cryptographic operations are computationally expensive, they are typically limited to this initial stage of the protocol. The resulting shared secret is then used to derive symmetric encryption keys, for example those used by AES, which protect the actual data exchanged during the communication session because symmetric algorithms are significantly more efficient for protecting high-throughput data streams [42, 59]. In this way, communication protocols combine asymmetric cryptography for authentication and key establishment with symmetric cryptography for efficient protection of bulk data, ensuring a balance between strong security guarantees and practical system performance.

To better understand how these protocols are organized within communication systems, it is useful to consider the layered abstraction models used in network design. Figure 2.7 illustrates the relationship between the Open Systems Interconnection (OSI) and Transmission Control Protocol/Internet Protocol (TCP/IP) models, together with representative protocols and their corresponding protocol data units across different layers.

OSI		TSP/IP	Protocols	Protocol Data Unit
Layer 7	Presentation	Application	DHCP DNS HTTPS SMTP	Application Data
Layer 6	Application			
Layer 5	Session			
Layer 4	Transport	Transport	TCP, UDP	Segment(TCP) Datagram(UDP)
Layer 3	Network	Internet	IPv4, IPv6, IPsec	Packet
Layer 2	Data Link	Network Interface	Ethernet, MACsec	Frame
Layer 1	Physical		Optical Fiber, DSL	Bit/Symbol

Figure 2.7: OSI and TCP/IP models with representative protocols and protocol data units. Adapted from Figure 3.1 of [2], with permission from the author.

These layered models provide a structured view of how communication systems are organized. While the OSI model defines seven conceptual layers, practical Internet architectures typically follow the simplified TCP/IP model, which groups these functions into fewer layers. Each layer is responsible for specific operations and introduces its own protocol-specific processing and data structures. In the context of cryptography, different security mechanisms are applied at different layers, ranging from application-level protection (e.g., TLS) to network-layer security (IPsec) and link-layer protection (MACsec).

Secure communication protocols such as TLS and IPsec typically separate session establishment from continuous data protection, corresponding here to a control plane and a data plane. The control plane is responsible for establishing the cryptographic context of a communication session through functions such as authentication, key establishment, parameter negotiation, and rekeying. These operations commonly rely on asymmetric cryptography and occur relatively infrequently.

After session establishment, the data plane uses the resulting symmetric keys to protect ongoing traffic. Because packets or records must be processed continuously and often at high throughput, efficient symmetric mechanisms such as AES-GCM are used for confidentiality and integrity. This separation avoids applying computationally expensive asymmetric operations to every unit of application traffic and provides the protocol-level architectural baseline used throughout the following chapters.

After a shared secret has been established during the handshake phase, protocols must transform this secret into operational encryption keys used to protect the communication session. This task is performed through a structured key schedule based on KDFs. A KDF takes the initial shared secret produced during key exchange and expands it into multiple cryptographically independent keys that serve different purposes within the protocol. This step is necessary because secrets established through key exchanges may not be uniformly random or may leak partial information, making them unsuitable for direct use as encryption keys.

In modern protocols such as TLS 1.3, HMAC-based Extract-and-Expand Key Derivation Function (HKDF) is commonly used for this purpose. HKDF follows a two-stage process consisting of an *extract* phase and an *expand* phase. In the extract stage, the entropy from the Input Key Material (IKM) is concentrated into a Pseudorandom Key (PRK). In the expand stage, this PRK is used to generate output key material of the desired length, which is subsequently divided into distinct functional keys used within the protocol. These derived keys may include keys used to protect the remaining steps of the handshake as well as keys used for encrypting application data.

A similar approach appears in other protocols such as Internet Key Exchange version 2 (IKEv2), where the shared secret produced during key exchange is combined with nonces and processed through a pseudorandom function to derive multiple session keys used for encryption, integrity protection, and the establishment of additional security associations [60].

A prominent example of this architecture is TLS 1.3, which separates session establishment from subsequent record protection. In a typical certificate-authenticated ephemeral Diffie–Hellman handshake, the client and server exchange ephemeral key shares, while the server authenticates the handshake transcript using its digital certificate and a corresponding digital signature. Both parties then derive shared secret material from the ephemeral key exchange. The TLS 1.3 key schedule uses HKDF to derive the handshake and application

traffic secrets required for the connection. Once the handshake is complete, the record protocol uses the derived traffic keys with the negotiated AEAD algorithm, such as AES-GCM, to protect application data [50].

Similar architectural principles appear in network-layer security protocols. The IPsec framework protects traffic at the network layer by encapsulating Internet Protocol (IP) packets within cryptographically protected structures using mechanisms such as the Encapsulating Security Payload (ESP) [61, 62]. In IPsec, IKEv2 is responsible for negotiating cryptographic parameters and establishing shared keys between communicating peers [60]. Once these parameters are agreed upon, the ESP mechanism protects the data packets themselves using symmetric authenticated encryption algorithms such as AES-GCM [51]. IPsec can operate in different modes depending on the deployment scenario. In transport mode, only the payload of an IP packet is encrypted, whereas in tunnel mode the entire packet is encapsulated within a new encrypted packet [61, 62]. Figure 2.8 illustrates the structure of an IPsec packet protected using the ESP mechanism.

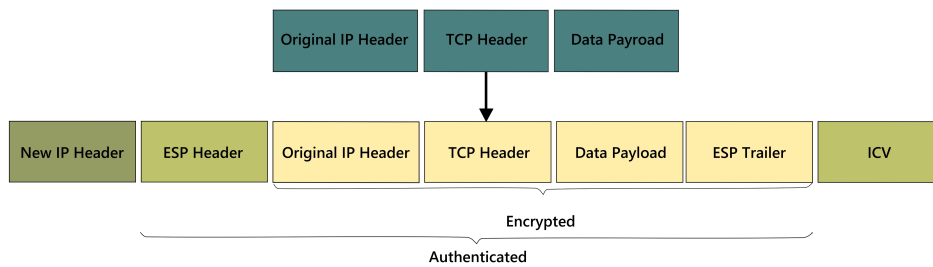


Figure 2.8: Structure of an IPsec ESP-protected packet. Adapted from Figure 5.1 of [1], with permission from the author.

In this construction, the original packet is encapsulated within an additional security layer that introduces headers and trailers that support encryption, integrity verification, and replay protection through sequence numbering. The payload is encrypted, while authentication data is appended to ensure that any modification of the packet can be detected. This design enables secure communication over untrusted networks while preserving compatibility with existing routing mechanisms.

A similar architectural approach is employed in MACsec, which protects Ethernet frames at the link layer [52]. MACsec adds a Security Tag (SecTAG) that enables replay protection. It also appends an Integrity Check Value (ICV) to the frame trailer in order to verify the integrity of the protected data [63]. Because the addressing information remains visible, frames can still be forwarded by network switches while their contents remain protected. Both IPsec and MACsec rely on dedicated key agreement mechanisms to manage the symmetric keys used for protecting data traffic, namely IKEv2 in the case of IPsec and MACsec Key Agreement (MKA) in MACsec deployments [60, 64]. Figure 2.9 illustrates the structure of an Ethernet frame protected using MACsec.

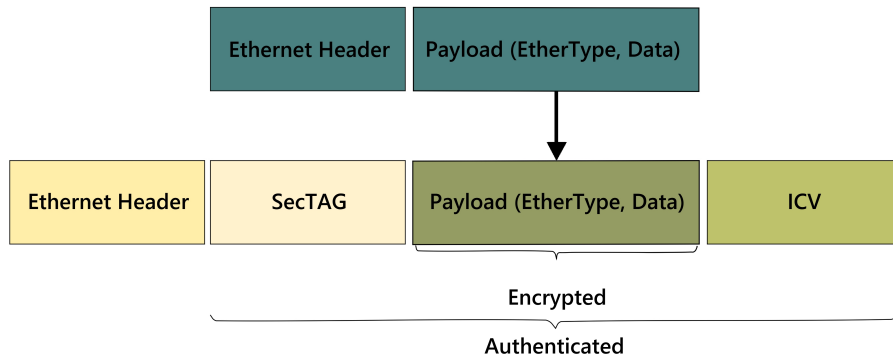


Figure 2.9: Structure of a MACsec-protected Ethernet frame. Adapted from Figure 6.1 of [1], with permission from the author.

In this setting, the original Ethernet frame is extended with dedicated security fields inserted between the Ethernet header and the payload. The SecTAG carries control information, including the packet number used by MACsec replay-protection processing [52], while the ICV ensures the authenticity and integrity of both the payload and the associated metadata. This design enables link-layer protection while preserving the addressing information necessary for frame forwarding.

Together, these protocol structures enabled authentication, key establishment, and efficient bulk data protection to scale across large digital infrastructures.

Because these mechanisms underpin authentication, secure communication, and trust relationships across distributed environments, cryptography effectively functions as a foundational component of modern digital infrastructure rather than as an isolated security feature.

These protocol architectures were developed under the assumption that the underlying cryptographic primitives would remain secure within the classical computational model described earlier. The following section examines why those assumptions remained stable for decades and how that stability shaped the evolution of modern digital infrastructures.

2.5 Stability of Classical Cryptographic Assumptions

As discussed in Section 2.1, classical cryptography relies on computational security rather than unconditional security, since perfect secrecy is impractical for large-scale digital communication. The long-term stability of this model depended on the persistence of a small number of hardness assumptions and on the ability of deployed systems to preserve security through parameter scaling.

Within the computational security framework introduced in Section 2.1, the reliability of classical cryptographic systems depends on a small set of mathematical problems that are believed to be computationally hard. The most prominent of these are the IFP and the DLP. Constructions such as RSA rely on the difficulty of recovering the prime factors of a large composite integer, while protocols such as DH and ECC rely on the hardness of computing discrete logarithms in suitable algebraic groups [41]. For sufficiently large parameters, no efficient classical algorithms are known that can solve these problems within realistic time frames. As a result, these problems function as trapdoor mechanisms because they are easy to compute in one direction but computationally difficult to invert without knowledge of the associated secret information. For several decades, these assumptions were widely regarded as secure against classical digital computers, allowing them to serve as the foundation for protecting sensitive data in government systems, financial infrastructures, and Internet communications [41, 42].

The continued effectiveness of these cryptographic constructions was largely due to the ability to maintain security through parameter scaling. For several decades, the dominant computational model remained classical, even as hardware became faster and more parallel. As computational capabilities improved, security levels could be preserved by increasing key sizes rather than redesigning cryptographic algorithms. This gradual evolution allowed the industry to migrate from RSA-1024 to RSA-2048, adopt larger elliptic-curve parameters, or increase symmetric key sizes from AES-128 to AES-256 when higher security margins were required. Because these hardness assumptions remained resistant to classical attacks, extending parameter sizes provided a practical way to maintain security against increasing computational power without fundamentally altering the cryptographic systems deployed across digital infrastructures.

Another factor contributing to this stability was the architectural model used in modern communication protocols. Secure communication systems typically adopted a hybrid design in which asymmetric cryptography was used only during the initial handshake to authenticate parties and establish shared secrets, while symmetric cryptography protected the bulk of transmitted data. This division of labor balanced security and performance by limiting expensive asymmetric operations to short-lived setup phases and relying on efficient symmetric encryption for sustained communication. Because these primitives offered predictable security and performance characteristics, they became standardized in widely deployed protocols such as TLS, IPsec, and MACsec and were gradually integrated into operating system network stacks and security services [59].

This long-standing reliability also enabled cryptographic protection to scale with the growth of digital infrastructure. Many cryptographic infrastructures are deployed for long operational lifetimes, meaning that systems designed today may remain active well beyond the point at which current cryptographic assumptions cease to hold. The widespread adoption of X.509 certificates, managed by globally trusted CAs, established a trust backbone that is included by default in most operating systems, browsers, and digital

devices [57]. At the same time, the efficiency of symmetric cryptography and its suitability for hardware acceleration allowed secure communication to keep pace with increasing network speeds and growing traffic volumes. As a result, cryptography came to function not as an optional add-on, but as a foundational component of modern digital infrastructure.

This stability depended fundamentally on the limits of classical computation. These assumptions are now being reconsidered due to the emergence of quantum computing, which represents a transformative shift in computational power [65]. Quantum algorithms, most notably Shor’s algorithm, can efficiently solve the integer factorization and discrete logarithm problems that underpin widely deployed asymmetric cryptographic systems such as RSA and ECC [28, 66]. If large-scale quantum computers become available, these algorithms would undermine the security foundations of many existing protocols. As a result, cryptographic schemes that currently secure global communication infrastructures could become vulnerable.

These developments challenge the long-standing assumptions that enabled classical cryptography to function as a stable foundation for digital infrastructure and motivate a coordinated transition toward PQC and quantum-resilient solutions [67]. The time-sensitive implications for long-lived data are examined in Chapter 3.

Chapter 3

The Quantum Threat Model

This chapter examines how the emergence of quantum computing alters the security assumptions underlying modern cryptographic systems. Building on the classical computational model established in Chapter 2, the present chapter analyzes how those assumptions change when adversaries are no longer limited to classical computation. It first explains how quantum computing affects the security model itself, then examines the impact of quantum algorithms on classical cryptographic primitives. The chapter subsequently discusses the consequences for protocols and infrastructures, the time-sensitive nature of the threat, and why classical cryptographic systems cannot be incrementally adapted to address it.

3.1 Quantum Computing and the Security Model

Modern cryptographic mechanisms were designed under the assumption that adversaries operate within the classical model of computation, as described in the previous chapter. Within this model, security is defined relative to the computational resources available to a classical digital computer. Many widely deployed cryptographic schemes rely on mathematical problems that are easy to compute in one direction but computationally infeasible to invert without secret information. For several decades, these assumptions proved stable because improvements in classical computing primarily increased speed and parallelism without altering the underlying computational model. As a result, security could often be preserved simply by increasing key sizes while retaining the same cryptographic constructions.

Quantum computing represents a fundamentally different approach to computation. Instead of relying on binary logic implemented through electronic circuits, quantum computers leverage physical phenomena such as superposition and entanglement to process information [68, 69, 70]. This change enables the development of quantum algorithms that can solve certain mathematical problems significantly more efficiently than classical algorithms. Consequently, the emergence of quantum computing is not simply a matter of faster com-

putation. It changes the computational model on which classical cryptographic security is based. The quantum threat therefore arises from a structural shift in adversarial capability, rather than from a gradual increase in processing power.

However, these security guarantees depend on the limits of classical computation. A change in the computational model therefore directly affects the hardness assumptions on which cryptographic systems rely. In particular, quantum algorithms challenge the hardness of problems such as integer factorization and discrete logarithms that underlie widely deployed public-key systems [28, 66].

The most significant consequence for cryptography is Shor’s algorithm, which shows that integer factorization and discrete logarithms can be solved efficiently on sufficiently powerful quantum computers [22]. Since these problems underpin systems such as RSA, DH, and ECC, large-scale quantum computation would undermine the public-key mechanisms used in modern PKI. The impact of such algorithms is examined in detail in the following section.

Although large-scale quantum computers are not yet generally available, their potential consequences make the threat relevant before such machines exist. This time-sensitive dimension of the threat, including HNDL scenarios, is discussed in Section 3.4.

From the perspective of cryptographic system design, the most important issue is not the physical implementation of quantum hardware, but the algorithmic implications of quantum computation for existing security assumptions. Whether quantum processors are built using superconducting circuits, trapped ions, or photonic systems, the critical factor for cryptography is the existence of algorithms that can solve previously intractable problems efficiently. This chapter therefore focuses on the security consequences of quantum computing for cryptographic infrastructures and communication protocols, rather than on quantum hardware architectures. The need for early planning and coordinated migration is already reflected in international cybersecurity guidance [30, 31, 67].

3.2 Quantum Attacks on Classical Cryptography

Quantum computing affects symmetric and asymmetric cryptographic systems in different ways. In classical settings, the security of these systems depends on the computational effort required to solve the underlying mathematical problems. The introduction of quantum algorithms alters this landscape by enabling certain problems to be solved more efficiently, thereby reducing the effective security of widely used cryptographic constructions.

Quantum computing affects symmetric cryptography primarily through Grover’s algorithm, which provides a quadratic speedup for exhaustive search attacks on unstructured spaces

such as key spaces or random oracles [71]. Classically, exhaustive key search requires

$$O(2^{|K|})$$

operations, where $|K|$ denotes the key length. Grover's algorithm reduces this complexity to

$$O(2^{|K|/2}),$$

thereby lowering the effort required to recover a secret key.

However, these asymptotic bounds should be interpreted carefully because they describe idealized query complexity rather than the full resource cost of a practical quantum attack. Practical attacks based on Grover's algorithm require reversible quantum circuits for the target primitive, such as AES. Such circuits may be deep and costly, and their implementation introduces additional challenges related to quantum noise and error correction [72, 73].

The practical consequence is often summarized as a reduction in the effective security level of symmetric ciphers. For example, AES-128 is sometimes described as offering roughly 64 bits of security against an idealized quantum adversary using Grover's algorithm. This statement should be interpreted as a simplified worst-case query-complexity estimate rather than as a direct practical attack cost. Practical Grover-based attacks require reversible implementations of the target primitive, long coherent computations, error correction, and substantial physical resources. Despite this reduction in the generic security margin, symmetric cryptography is not structurally broken. Bernstein emphasizes that Grover's algorithm has a substantially less disruptive effect on secret-key cryptography than Shor's algorithm has on classical public-key systems, and that its effect can be compensated by choosing larger key sizes [34]. Accordingly, higher security margins can be obtained by increasing symmetric-key sizes, for instance by using AES-256, which is commonly treated as providing an adequate post-quantum symmetric security margin [34, 72].

A similar effect applies to cryptographic hash functions, although the impact depends on the specific security property considered. For preimage resistance, Grover's algorithm reduces the computational effort required to invert a hash function from $O(2^n)$ to approximately $O(2^{n/2})$, where n denotes the output length [71]. For collision resistance, the Brassard–Høyer–Tapp (BHT) algorithm reduces generic collision search to approximately $O(2^{n/3})$ quantum queries in the standard query model [74]. Although these results do not render hash functions insecure in the same structural sense as Shor's algorithm affects public-key cryptography, they influence the required output lengths and security margins for hash-based constructions in quantum settings.

In contrast, quantum computing poses a structural threat to asymmetric cryptography through Shor's algorithm [22]. Unlike the quadratic speedup associated with Grover's algorithm, Shor's algorithm provides an exponential improvement over the best-known

classical algorithms for solving the IFP and the DLP. These problems form the mathematical foundation of widely deployed public-key systems, including RSA, DH, and ECC. If sufficiently powerful quantum computers become available, the trapdoor assumptions that guarantee the security of these schemes no longer hold. As a result, the security foundations of classical PKI are fundamentally compromised.

The impact of quantum computing is therefore fundamentally different for symmetric and asymmetric cryptography. For symmetric schemes, the effect is quantitative because the quadratic speedup provided by Grover’s algorithm reduces the effective security level. This reduction is significant but manageable and can be countered by increasing the key length [72]. In contrast, for asymmetric cryptography, the effect is structural because the mathematical problems that provide security can be efficiently solved using quantum algorithms, making classical constructions insecure regardless of how large the parameters are. While symmetric cryptography remains usable through parameter adjustments, asymmetric cryptography requires a transition to entirely new mathematical frameworks, commonly referred to as PQC.

Based on these results, a clear distinction can be made between cryptographic primitives that are structurally broken and those that remain usable with reduced security. Public-key mechanisms such as RSA, DH, and ECC, including signature schemes such as Elliptic-Curve Digital Signature Algorithm (ECDSA) and Edwards-curve Digital Signature Algorithm (EdDSA), become insecure in the presence of quantum adversaries [75]. In contrast, symmetric primitives such as block ciphers and cryptographic hash functions are not directly broken, but their security levels are reduced due to quantum search algorithms. Consequently, maintaining adequate security requires the use of larger key sizes and longer hash outputs.

Table 3.1: Proposed transition milestones for selected quantum-vulnerable public-key mechanisms in the NIST IR 8547 initial public draft. Adapted from Table 2.2 of [2], with permission from the author; milestones updated from [75].

Mechanism	Security strength	Deprecated after	Disallowed after
Finite-field DH	112 bits	2030	2035
Finite-field DH	≥ 128 bits	–	2035
Elliptic-curve DH	112 bits	2030	2035
Elliptic-curve DH	≥ 128 bits	–	2035
RSA	112 bits	2030	2035
RSA	≥ 128 bits	–	2035
ECDSA	112 bits	2030	2035
ECDSA	≥ 128 bits	–	2035
EdDSA	≥ 128 bits	–	2035

Table 3.1 summarizes selected classical public-key mechanisms that are vulnerable in the presence of a sufficiently capable quantum adversary. These mechanisms are used for key establishment and digital signatures and rely on mathematical problems that can be solved efficiently by Shor’s algorithm. The table also shows the proposed deprecation and disallowance milestones in the initial public draft of NIST IR 8547, which provide migration-planning reference points rather than estimates of when cryptographically relevant quantum computers will become available [75].

Application-specific or national guidance may require earlier transition, particularly for key-establishment mechanisms that expose long-lived confidential data to harvest-now-decrypt-later risks; broader migration guidance is discussed in Chapter 4.

These changes have direct implications for practical security levels. The notion of “bits of security” must be reconsidered in the presence of quantum adversaries, especially because quantum search changes the cost model used to compare symmetric and post-quantum mechanisms [75]. Rather than treating the reduction in symmetric security as a precise practical attack estimate, post-quantum standardization uses established symmetric primitives as reference benchmarks for security categories. This allows post-quantum schemes to be compared against familiar targets such as AES-128, AES-192, and AES-256.

To provide a standardized reference for evaluating post-quantum mechanisms, NIST defines security categories based on the computational effort required to break well-established symmetric primitives [75]. These categories provide practical benchmarks for parameter selection, as shown in Table 3.2.

Table 3.2: NIST post-quantum security categories. Adapted from Table 2.1 of [2], with permission from the author; security categories from [75].

Security Level	Attack Type	Example
1	Key search block cipher with a 128-bit key	AES-128
2	Collision search on a 256-bit hash function	SHA-256
3	Key search block cipher with a 192-bit key	AES-192
4	Collision search on a 384-bit hash function	SHA-3-384
5	Key search block cipher with a 256-bit key	AES-256

These categories align post-quantum parameter sets with established security targets, such as those associated with AES-128, AES-192, and AES-256. They therefore provide a system-oriented basis for comparing post-quantum schemes under classical and quantum attack models.

3.3 What Quantum Adversaries Change in Systems

The impact of quantum adversaries extends beyond individual cryptographic primitives and affects the fundamental mechanisms through which trust is established in digital systems. PKI serves as the backbone of identity and trust on the Internet. It is built on digital certificates and hierarchical chains of trust, as illustrated in Figure 3.1 [57].

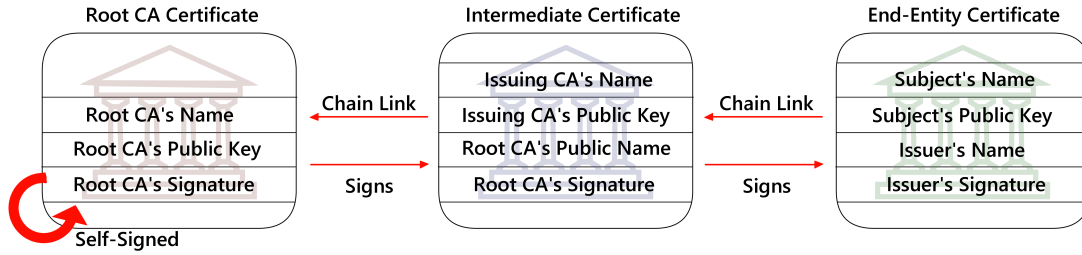


Figure 3.1: Certificate-based chain of trust in PKIs. Adapted from Figure 1 of [10], licensed under CC BY 4.0.

In this model, trust originates from a root CA, which serves as the anchor of trust, while intermediate certificates extend this trust toward end-entity certificates used by applications. Each step in this chain is validated through digital signatures, allowing entities to verify identities without requiring direct prior trust relationships. When certificate chains rely on quantum-vulnerable signature algorithms based on assumptions such as integer factorization or discrete logarithms, a sufficiently capable quantum adversary could compromise the signatures used to establish and propagate trust. Consequently, certificate-based trust models that continue to rely on such algorithms would no longer provide reliable authentication guarantees. This would affect services that depend on certificate-based identity, including secure web communication, software distribution, and machine authentication.

This breakdown of trust directly affects authentication mechanisms, which depend on digital signatures to verify identity and ensure message integrity. In modern systems, authentication enables entities to confirm the origin of data and interact securely without prior shared secrets. However, when the underlying signature schemes become vulnerable, adversaries can forge identities or impersonate legitimate parties during communication. As a result, authentication can no longer be assumed to be reliable in the presence of quantum-capable adversaries, weakening the ability to establish trustworthy communication channels.

Classical DH and ECDH mechanisms are widely used for key establishment in protocols such as TLS and IKEv2, while MACsec relies on its own key-management framework and may depend on classical trust or key-establishment mechanisms elsewhere in the infrastructure. When the underlying mathematical problems become solvable by quantum

adversaries, the confidentiality of the key establishment process is no longer guaranteed. An adversary capable of recovering the exchanged secret can derive the session keys used to protect subsequent data transmission.

Within the control-plane/data-plane structure introduced in Chapter 2, the quantum threat is concentrated primarily in authentication and key establishment because these functions depend on asymmetric cryptographic assumptions vulnerable to quantum attacks. Symmetric data protection remains structurally usable when appropriate parameters are selected. Nevertheless, compromise of the session-establishment process can expose the keys used for subsequent communication, undermining the security of the entire session even when the data-protection primitive itself remains computationally strong.

These effects propagate across entire systems because cryptographic functions are embedded in protocols, operating-system network stacks, cloud platforms, and distributed services. As secure communication has become a default requirement, vulnerabilities in foundational cryptographic assumptions affect not only individual applications but also the infrastructure in which they operate. Infrastructure therefore amplifies cryptographic change, since modifications to trust establishment and key exchange propagate through protocols, software stacks, and operational workflows.

Addressing these challenges requires more than replacing vulnerable algorithms. Cryptographic functions are tightly integrated into protocol formats, certificate ecosystems, software libraries, and hardware-supported execution paths. Migration must therefore preserve interoperability and performance while changing the assumptions on which authentication and key establishment depend.

3.4 The Urgency of the Quantum Threat

The security of modern communication systems must be evaluated not only in terms of current protection but also in relation to future computational capabilities. Classical cryptographic mechanisms have provided reliable security under the assumption of bounded computational resources. However, this assumption may not remain valid indefinitely. As a result, the relevance of the quantum threat extends beyond the point at which large-scale quantum computers become available. Even before such systems are deployed, the possibility of future advances creates uncertainty in the long-term security of today's cryptographic infrastructure.

The HNDL paradigm describes a scenario in which an adversary records and stores encrypted traffic today with the intent of decrypting it in the future once a sufficiently powerful quantum computer becomes available [29]. This threat is particularly relevant for key exchange mechanisms, which can be broken retroactively if their underlying public-key assumptions fail. Even when bulk data is protected using strong symmetric encryption, the

session keys may have been established using classical mechanisms such as RSA, DH, or ECDH. Consequently, confidential information exchanged today may already be exposed to future risk.

The severity of this risk depends on the required confidentiality period of the data. Some information must remain confidential for many years or decades. If that protection period extends beyond the time required for cryptographic migration and the possible arrival of cryptographically relevant quantum capability, ciphertext recorded today may become vulnerable to retrospective decryption [30, 31].

This leads to a fundamental mismatch between cryptographic lifetime and data lifetime. Cryptographic lifetime denotes the period during which a given cryptographic scheme is expected to remain secure under current computational assumptions. In contrast, data lifetime refers to the duration for which the protected information must remain confidential. Historically, these lifetimes were easier to align because advances in classical computation could often be addressed by increasing key sizes [42]. Quantum computation disrupts this relationship by changing the computational assumptions themselves. If a cryptographic scheme is expected to become insecure within a certain timeframe, but the data it protects must remain confidential for a longer period, then the data is likely to be exposed once the underlying assumptions are no longer valid.

The notion of “Q-day” is often used to describe the point at which quantum computers become capable of breaking widely deployed public-key cryptographic systems [76]. While the exact timing of this transition remains uncertain, it represents a critical threshold beyond which the security guarantees of current infrastructures can no longer be relied upon. This uncertainty complicates security planning, as systems deployed today may need to remain secure over time periods that extend beyond this transition.

This uncertainty is reflected in migration guidance for quantum-vulnerable public-key mechanisms. Algorithms such as finite-field DH, ECDH, RSA, ECDSA, and EdDSA are identified as requiring replacement in the post-quantum transition. Migration is therefore a planning problem involving algorithm deprecation, certificate replacement, protocol interoperability, and long deployment lifetimes.

These considerations create a strong incentive for early action. Cryptographic systems are deeply embedded in protocols, software, and hardware environments. Replacing them therefore requires coordinated effort across multiple layers of the technology stack. Because migration processes can take many years to complete, delaying the transition increases the volume of sensitive data that may later be exposed to decryption. For this reason, the quantum threat is not solely a future concern but a present-day challenge that requires proactive planning to ensure long-term confidentiality and resilience [30, 67].

3.5 Why Classical Cryptography Cannot Be Incrementally Fixed

Historically, cryptographic systems adapted to increasing computational power by scaling key sizes while preserving the underlying algorithms. Transitions such as RSA-1024 to RSA-2048 allowed systems to maintain security as classical hardware improved. This approach worked because the computational model remained unchanged. Quantum computing breaks this pattern. Algorithms such as Shor’s algorithm do not merely reduce the security margin of classical public-key systems. Instead, they invalidate the mathematical assumptions on which those systems rely. As a result, the post-quantum transition cannot be understood as a simple parameter update.

This limitation reflects a deeper difference between symmetric and asymmetric cryptography under the quantum model. Symmetric cryptography is affected by the quadratic speedup of Grover’s algorithm, which can be mitigated by increasing key sizes [71]. Classical asymmetric cryptography is affected differently. The security of widely deployed systems such as RSA and ECC depends on the hardness of integer factorization and discrete logarithms, and Shor’s algorithm can solve these problems efficiently on sufficiently powerful quantum computers [22]. Even substantial increases in parameter sizes, such as moving to RSA-4096, do not restore the underlying security assumption. Consequently, the mathematical trapdoors underlying PKI no longer provide a reliable basis for secure communication.

The need to replace these primitives is further complicated by the fact that cryptographic mechanisms are deeply embedded within communication protocols and system architectures. As a result, transitioning to PQC is not a simple drop-in replacement of algorithms. Post-quantum schemes introduce larger keys, signatures, and ciphertexts, as well as different computational profiles [36, 37]. These larger cryptographic artifacts can increase bandwidth requirements, contribute to packet fragmentation, and introduce additional processing overhead during protocol handshakes, certificate validation, and session establishment [9].

One visible consequence appears in certificate infrastructures, where post-quantum migration requires changes to certificate contents and validation logic. Existing trust infrastructures cannot therefore be secured simply by increasing classical key sizes; certificate formats, signature mechanisms, and validation procedures must also evolve. Hybrid certificate approaches that support this transition are discussed in Chapter 4.

In addition to technical integration challenges, this transition must also consider backward compatibility and operational constraints. Many existing systems rely on long-established cryptographic standards and cannot be replaced abruptly without disrupting their functionality. As a result, migration strategies often rely on hybrid approaches that combine classical and post-quantum mechanisms within the same protocol execution. While these approaches enable a gradual transition, they introduce additional complexity and overhead.

Furthermore, infrastructures such as certificate ecosystems and trust hierarchies involve multiple interacting entities, making coordinated upgrades difficult. These constraints highlight that migration extends beyond a purely technical problem and becomes an organizational and ecosystem-level challenge.

Chapter 4

Post-Quantum Cryptography in Transition

This chapter examines how cryptographic systems adapt to the challenges introduced by quantum computing. Building on the analysis of quantum threats presented in Chapter 3, the focus shifts toward the mechanisms designed to address these challenges and the implications of their adoption. It introduces the mathematical foundations of post-quantum cryptography and the problem families that support quantum-resistant constructions, while outlining how these differ from classical approaches. The discussion continues with an examination of the trade-offs that characterize these schemes, emphasizing the balance between performance, security, and implementation complexity. It then turns to hybrid cryptographic mechanisms as a practical path for gradual adoption, followed by a discussion of the migration process toward quantum-resilient infrastructures. The chapter concludes by examining the broader system-level consequences of this transition, showing that the adoption of post-quantum cryptography extends beyond algorithmic replacement and influences the design of modern computing environments.

Source note. The mathematical foundations, algorithm trade-offs, and parts of the hybrid-cryptography discussion in this chapter draw substantially on Chapter 3 of [1]. Selected background material on quantum-vulnerable cryptographic standards also draws on Chapter 2 of [2]. The discussion has been updated and extended using the standards, migration guidance, implementation-security literature, and other sources cited throughout.

4.1 Post-Quantum Cryptography and New Foundations

As shown in Chapter 3, quantum algorithms undermine the hardness assumptions behind widely deployed public-key mechanisms such as RSA, DH, and ECC. Post-quantum cryptography addresses this problem by replacing classical number-theoretic assumptions

NIST Cryptography Standards			
Asymmetric Key Signature (FIPS 186) Key Establishment (800-56A/B/C) Subject to Quantum Threat	Symmetric Key	Guidelines	Tools
	AES (FIPS 197) TDEA (800-67)	Hash usage/security (800-107)	RNG (800-90A/B/C) KDF (800-108, 800-135)
	SHA-1/2 (FIPS 180) SHA-3 (FIPS 202)	Key Generation (800-133)	
	HMAC (FIPS 198) Randomized hash (800-106)	Key Management (800-57)	

Figure 4.1: NIST cryptographic standards and their exposure to quantum threats. Adapted from Figure 2.1 of [2], with permission from the author.

with alternative mathematical foundations that are believed to resist both classical and quantum attacks.

Figure 4.1 provides an overview of the cryptographic standards defined by NIST, organized across asymmetric and symmetric primitives, as well as supporting guidelines and tools. It also illustrates the asymmetric impact of the quantum threat. Public-key mechanisms require replacement, while symmetric primitives generally remain usable with larger parameters.

To address this shift, PQC replaces classical number-theoretic assumptions with alternative mathematical problems that are believed to remain hard for both classical and quantum adversaries. Instead of relying on integer factorization or discrete logarithms, post-quantum schemes are based on different problem families, including lattice-based cryptography, code-based cryptography, hash-based signatures, and multivariate constructions.

Among these families, lattice-based cryptography has become especially prominent. It forms the basis of standardized mechanisms such as ML-KEM, based on CRYSTALS-Kyber, and ML-DSA, based on CRYSTALS-Dilithium [77, 78]. Hash-based schemes include SLH-DSA, based on SPHINCS+, which relies on the security of cryptographic hash functions and provides conservative security guarantees, although typically with larger signatures and lower performance [79]. Code-based approaches rely on decoding problems in error-correcting codes and provide a mathematically distinct alternative to lattice-based constructions, although they often involve different key-size and performance trade-offs [80]. Multivariate constructions are based on the difficulty of solving systems of multivariate polynomial equations, but several candidates in this family have faced significant cryptanalytic challenges. The practical break of Rainbow provides a prominent example and illustrates why continued cryptanalysis, cautious deployment, and algorithmic diversity remain important in the post-quantum setting [81, 82].

Hash-based signatures also include stateful schemes such as the eXtended Merkle Signature Scheme (XMSS) and the Leighton-Micali Signature system (LMS). These schemes are approved for use in NIST SP 800-208, which specifies recommended parameter sets for stateful hash-based signatures, and they rely on the security of hash functions rather than lattice- or code-based assumptions [83]. Their main limitation is operational rather than mathematical. Signing states must be managed carefully, since reusing one-time signing material can compromise security. For this reason, XMSS and LMS are most suitable for controlled environments where state management can be enforced reliably, while SLH-DSA provides a stateless alternative with larger signatures and lower performance.

The role of Hamming Quasi-Cyclic (HQC) is particularly important for algorithmic diversity. HQC is a code-based KEM whose security is based on decoding problems for structured error-correcting codes, providing a mathematical foundation distinct from the structured-lattice assumptions used by ML-KEM [84]. In 2025, NIST selected HQC for standardization as an additional post-quantum KEM and as a backup to ML-KEM, which remains the primary standardized choice for general-purpose key establishment [84]. The purpose of this second KEM is therefore not to replace ML-KEM, but to diversify the underlying security assumptions so that a future cryptanalytic weakness affecting structured lattices does not leave the post-quantum key-establishment portfolio dependent on a single problem family. This diversification comes with different implementation, computational, and communication trade-offs, so the suitability of HQC remains deployment-dependent.

Together, these families define a diverse post-quantum design space. Each offers different trade-offs in terms of security maturity, computational cost, key and signature sizes, and implementation complexity. For this reason, lattice-based cryptography is used in the following discussion as a representative example of how post-quantum security is constructed in practice.

In addition to these mathematical foundations, the transition toward PQC is driven by standardization. The NIST post-quantum standardization process led to the finalization of ML-KEM as a key encapsulation mechanism in Federal Information Processing Standards (FIPS) 203, ML-DSA as a digital signature algorithm in FIPS 204, and SLH-DSA as a stateless hash-based digital signature algorithm in FIPS 205 [36, 37, 38]. Falcon was also selected for standardization and is being developed by NIST under the name FFT over NTRU-Lattice-Based Digital Signature Algorithm (FN-DSA), while HQC was selected in 2025 as an additional code-based key encapsulation mechanism [35, 84]. These developments show that post-quantum standardization is not limited to a single mathematical family, but instead aims to preserve algorithmic diversity and deployment flexibility.

Throughout the remainder of the book, the standardized names ML-KEM, ML-DSA, and SLH-DSA are used for the finalized NIST algorithms. Former submission names are retained only when referring specifically to pre-standardization implementations, historical

measurements, vulnerability names, or source-specific terminology. Falcon is treated separately because its NIST standardization as FN-DSA remains under development.

Unlike classical cryptographic assumptions such as integer factorization and discrete logarithms, which have been studied for several decades, many post-quantum constructions rely on relatively newer problem formulations. Although no efficient classical or quantum attacks are currently known for these problems, their long-term security remains an active area of research. This uncertainty motivates a cautious deployment approach and reinforces the need for multiple algorithmic solutions, each offering different trade-offs in terms of performance, key sizes, and implementation complexity.

To better understand how security is achieved in these schemes, it is useful to introduce the main mathematical objects used in lattice-based cryptography. A lattice $\mathcal{L}$ is a discrete additive subgroup of $\mathbb{R}^n$. Equivalently, given linearly independent basis vectors $\mathbf{b}_1, \dots, \mathbf{b}_n \in \mathbb{R}^n$, the lattice generated by this basis is defined as

$$\mathcal{L}(\mathbf{b}_1, \dots, \mathbf{b}_n) = \left\{ \sum_{i=1}^n a_i \mathbf{b}_i \mid a_i \in \mathbb{Z} \right\} = \{B\mathbf{a} \mid \mathbf{a} \in \mathbb{Z}^n\}.$$

The same lattice may be represented by different bases, and this distinction between easy and hard representations underlies many lattice-based constructions [1].

Figure 4.2 illustrates a simple two-dimensional lattice generated by the basis matrix

$$B = \begin{bmatrix} 2 & 0 \\ 1 & 3 \end{bmatrix}.$$

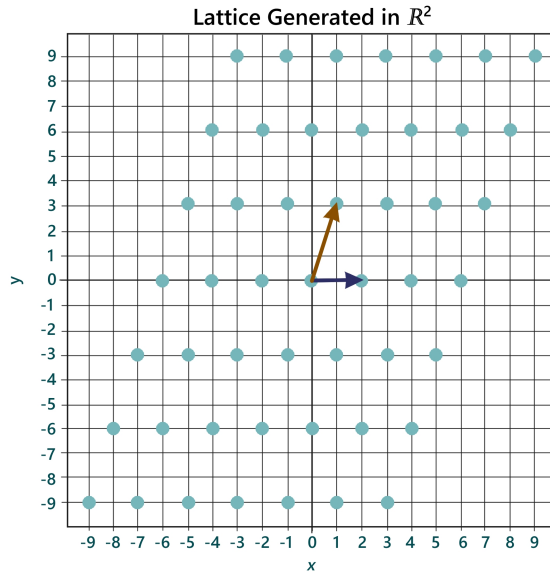


Figure 4.2: Two-dimensional lattice generated by the basis matrix B . Adapted from Figure 3.1 of [1], with permission from the author.

Although this example is low-dimensional, practical lattice-based schemes operate over high-dimensional algebraic structures where recovering hidden short vectors or solving noisy linear relations is believed to be infeasible for both classical and quantum adversaries.

One central hardness assumption is the LWE problem. Let $n \geq 1$ and $q \geq 2$ be integers, let χ be an error distribution over $\mathbb{Z}_q$, and let $\mathbf{s} \in \mathbb{Z}_q^n$ be a secret vector. The LWE distribution samples

$$\mathbf{a} \leftarrow \mathbb{Z}_q^n, \quad e \leftarrow \chi,$$

and outputs pairs of the form

$$(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q,$$

where all operations are performed modulo q [1, 77]. The search version asks an adversary to recover $\mathbf{s}$ from multiple such samples, while the decision version asks the adversary to distinguish these samples from uniformly random elements of $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

For efficiency and compactness, practical schemes often use structured variants of this problem. The MLWE problem generalizes LWE to module settings over richer algebraic rings. Let R be the ring of integers of a number field, and let $R_q = R/qR$ denote its quotient modulo q . Let $d \geq 1$ be the module rank, and let χ be an error distribution over R_q . For a secret $\mathbf{s} \in R_q^d$, the MLWE distribution samples

$$\mathbf{a} \leftarrow R_q^d, \quad e \leftarrow \chi,$$

and outputs

$$(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e).$$

This structure allows schemes such as ML-KEM and ML-DSA to obtain practical performance while preserving hardness assumptions related to lattice problems [77, 78].

A related assumption used in lattice-based signatures is the SIS problem. Given a matrix $A \in \mathbb{Z}_q^{n \times m}$, the goal is to find a nonzero vector $\mathbf{z} \in \mathbb{Z}^m$ such that

$$\|\mathbf{z}\| \leq \beta \quad \text{and} \quad A\mathbf{z} \equiv 0 \pmod{q}.$$

In structured variants, this problem is also defined over polynomial rings. SIS-type assumptions are central to lattice-based digital signatures, where security depends on the difficulty of producing short valid responses that satisfy public verification equations [1, 78].

An important practical aspect of these constructions is their reliance on large amounts of pseudorandom material. In lattice-based schemes, operations such as sampling noise vectors, generating structured matrices, and deriving intermediate values require outputs that exceed the size of traditional hash digests. For this reason, many post-quantum algorithms rely on

XOFs, such as those defined in the SHA-3 standard. These functions enable the generation of arbitrarily long pseudorandom sequences from compact seeds, ensuring both efficiency and reproducibility. As a result, XOFs play a central role in the implementation of modern post-quantum schemes, particularly in lattice-based constructions such as ML-KEM and ML-DSA, where they are used to deterministically generate public matrices from compact seeds and to sample noise distributions required for security [36, 37, 54].

No polynomial-time quantum algorithms analogous to Shor’s algorithm are known for the principal lattice-based assumptions used by current post-quantum standards or for the code-based assumptions considered in ongoing standardization efforts. Quantum algorithms can nevertheless affect concrete security estimates, so parameter selection accounts for the best-known classical and quantum attacks rather than assuming that these problems are unaffected by quantum computation [35, 36, 37, 84].

As discussed in the context of classical cryptography, modern communication systems continue to rely on two core asymmetric functions, key establishment and authentication. These roles do not disappear in the post-quantum setting, but their underlying constructions change fundamentally. In many current post-quantum migration profiles, classical DH-style key exchange is replaced or combined with KEMs, while classical signature mechanisms are replaced or combined with post-quantum signature algorithms.

This shift is not only a matter of design preference but also reflects structural differences in the underlying mathematics. Classical KEX protocols rely on algebraic properties that allow both parties to contribute to the derivation of a shared secret. In contrast, many post-quantum constructions are naturally expressed as key encapsulation mechanisms, in which one party uses the recipient’s public key to produce a ciphertext and shared secret, while the recipient derives the corresponding secret through decapsulation using the private key.

In practical terms, this transition is reflected in the use of schemes such as ML-KEM for shared-secret establishment and ML-DSA for authentication. The significance of this shift lies not in a change of cryptographic purpose, but in a change of mathematical foundation, as these primitives are no longer built on integer factorization or discrete logarithms, but on problem families believed to remain hard even for quantum adversaries.

In practical deployments, a KEM must remain secure against active adversaries that can modify ciphertexts and interact with the decapsulation process. For this reason, ML-KEM combines an underlying MLWE-based encryption scheme with the Fujisaki–Okamoto Transformation (FO) transformation, which converts an Indistinguishability under Chosen-Plaintext Attack (IND-CPA)-secure construction into an Indistinguishability under Chosen-Ciphertext Attack (IND-CCA)-secure KEM. This provides robustness against adaptive attacks in protocol environments [36, 77].

At a high level, ML-DSA follows a structure analogous to Schnorr-type signature schemes. The signer generates a commitment, derives a challenge by hashing the message together with this commitment, and computes a response that proves knowledge of the secret key without revealing it. In lattice-based settings, additional care is required to prevent information leakage through the response distribution. For this reason, techniques such as rejection sampling are employed to ensure that signatures follow a suitable statistical distribution. While effective, this introduces variable iteration counts and implementation complexity that require careful side-channel analysis, which must be carefully addressed in practice [37, 78].

At the system level, post-quantum cryptographic schemes introduce significant differences. One of the most immediate consequences is their increased spatial footprint. Public keys, private keys, and signatures are often substantially larger than their classical counterparts, directly affecting bandwidth usage and protocol efficiency. Their computational profiles also differ substantially from classical mechanisms and depend on the scheme, operation, parameter set, implementation, and hardware platform. Polynomial arithmetic, matrix-vector operations, structured transforms such as the Number-Theoretic Transform (NTT), and pseudorandom generation through XOFs are important computational kernels in lattice-based schemes. Their relative cost depends on the algorithm, implementation, operation, and hardware platform [15, 85].

Overall, post-quantum schemes introduce a different design space from classical public-key cryptography. They rely on new hardness assumptions, use different algorithmic structures, and produce different computational and communication profiles. These differences make algorithm selection dependent on the constraints of the target system, which motivates the trade-off analysis in the next section.

4.2 Trade-offs in Post-Quantum Cryptography

The transition toward PQC introduces a more diverse design space than classical public-key cryptography. Post-quantum schemes are based on different mathematical problem families, including lattice-based, hash-based, code-based, and multivariate constructions, each with distinct implications for performance, artifact size, implementation complexity, and security maturity. Algorithm selection therefore becomes a multi-dimensional optimization problem rather than a choice among interchangeable primitives.

This variety of approaches is not only a result of mathematical diversity, but also a deliberate design choice. From a security perspective, relying on a single computational assumption would introduce systemic risk, since a breakthrough in one problem domain could compromise the entire infrastructure. For this reason, standardization efforts have focused on selecting multiple algorithms rather than converging on a single primitive [35]. From a deployment perspective, this enables different schemes to be deployed depending

on the requirements of the target environment, while also supporting hybrid constructions that combine multiple cryptographic mechanisms to provide resilience during the transition period. Figure 4.3 provides a comparative view of the main trade-offs among selected post-quantum cryptographic schemes. It illustrates the relationships between public key sizes, private key sizes, signatures, and ciphertexts across both signature schemes and key encapsulation mechanisms. These dimensions reflect the fundamental design tensions discussed above, where improvements in one metric often lead to increased costs in another.

In particular, the figure highlights that post-quantum schemes do not cluster around a single balanced region, but instead occupy distinct regions of the design space. This reflects the fact that different mathematical constructions inherently favor different performance characteristics, reinforcing the need to match algorithms to specific deployment contexts rather than treating them as interchangeable components.

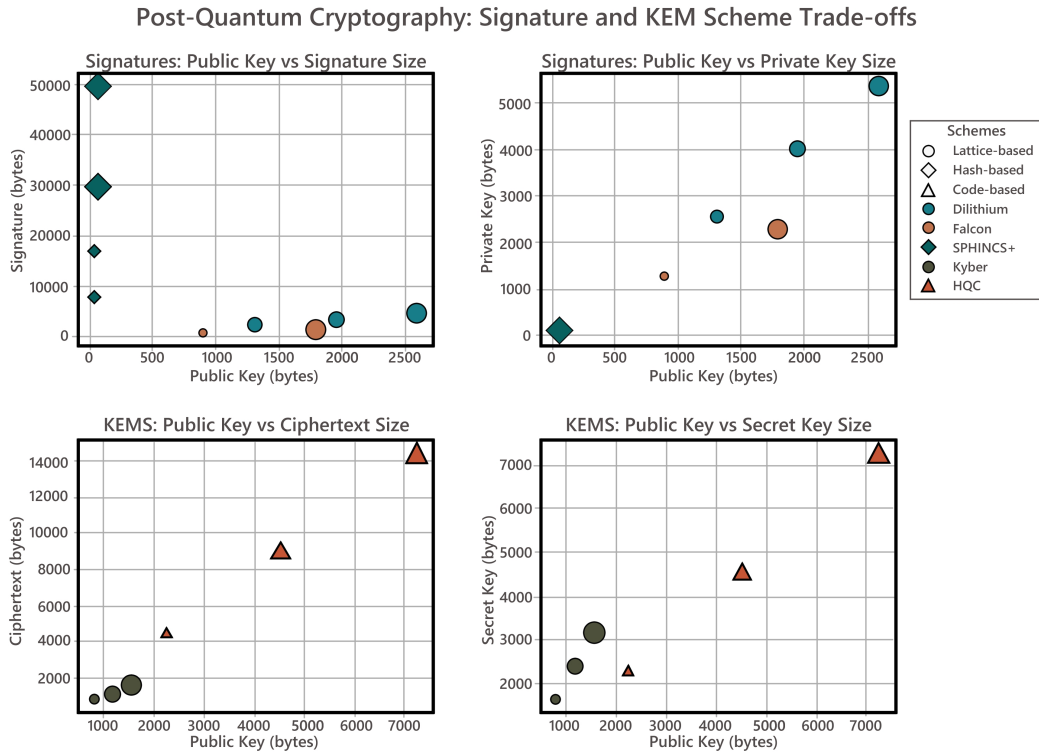


Figure 4.3: Trade-offs among selected post-quantum cryptographic algorithms. Adapted from Figure 3.6 of [1], with permission from the author.

Building on the comparison shown in Figure 4.3, a key trade-off concerns the relationship between execution speed and spatial footprint. Lattice-based schemes such as ML-DSA offer high execution speeds for both signing and verification, making them suitable for high-throughput systems. However, this efficiency comes at the cost of substantially larger public keys and signatures compared to classical schemes. In contrast, FN-DSA provides

more compact signature sizes and efficient verification, but its key generation process is computationally demanding and more difficult to implement efficiently.

In particular, Falcon implementations use complex Gaussian sampling techniques, and prominent implementations employ floating-point arithmetic, creating additional constant-time and side-channel engineering challenges [86].

Implementation security forms another important trade-off. Post-quantum schemes may be secure at the mathematical level while remaining vulnerable in practice to timing, power, electromagnetic, or cache side channels, as well as fault-injection attacks, if their implementations do not adequately control observable or fault-dependent behavior [87]. Lattice-based implementations contain several operations that require particular care, including sampling, rejection sampling, polynomial arithmetic, compression, and decapsulation checks. KyberSlash demonstrated that secret-dependent division operations in implementations of CRYSTALS-Kyber, the submission underlying ML-KEM, could produce exploitable timing leakage [88]. Separately, CVE-2024-37880 documented a timing vulnerability in a pre-standardization Kyber reference implementation in which certain LLVM Clang compilation configurations could introduce secret-dependent control flow, enabling secret-key recovery under the affected conditions [89]. These examples show that post-quantum security depends not only on the hardness of the underlying mathematical problem, but also on constant-time implementation, compiler-aware validation, careful randomness handling, side-channel testing, and secure integration on the target platform.

These differences illustrate how performance gains in one dimension often lead to increased costs in another, making it necessary to prioritize based on application requirements [6].

A second important dimension is the trade-off between security maturity and performance. Hash-based schemes such as SLH-DSA rely on well-understood cryptographic hash functions, providing strong confidence in their long-term security. However, this robustness comes at the cost of considerably reduced performance and very large signature sizes, which can reach tens of kilobytes and be challenging for latency- and bandwidth-sensitive protocols because of their substantially larger signatures and higher computational cost [79, 90]. In contrast, lattice-based constructions offer substantially better performance, but their underlying assumptions have been studied for a shorter period of time.

Beyond performance and security considerations, post-quantum algorithms also differ in how computational workloads are distributed across communicating parties. In a typical authenticated client-server setting, the distribution of work depends on which endpoint performs signing and which endpoint performs verification. The ML-DSA/ML-KEM stack provides a comparatively balanced computational profile across signing, verification, encapsulation, and decapsulation. In contrast, the FN-DSA/ML-KEM stack offers compact signatures and efficient verification, but signing and key generation are more computationally demanding. This means that FN-DSA-based deployments may reduce client-side cost when the client primarily verifies server signatures, but may increase server-side cost when the

server performs frequent signing operations. The practical advantage therefore depends on the protocol role, authentication pattern, and deployment environment [6].

To further illustrate these differences, Figure 4.4 compares the computational cost of the main operations in ML-DSA and FN-DSA, namely key generation, signing, and verification. The figure highlights that the performance characteristics of these schemes are not uniform across all operations, but instead depend strongly on how computation is distributed between client and server.

The following benchmark results were obtained using the pre-standardization CRYSTALS-Dilithium, Falcon, and CRYSTALS-Kyber implementations evaluated in [6]. These algorithms form the respective lineages of ML-DSA, the developing FN-DSA, and ML-KEM, but the original algorithm names are retained when reporting the measured results in order to distinguish the evaluated implementations from the finalized or still-developing NIST specifications.

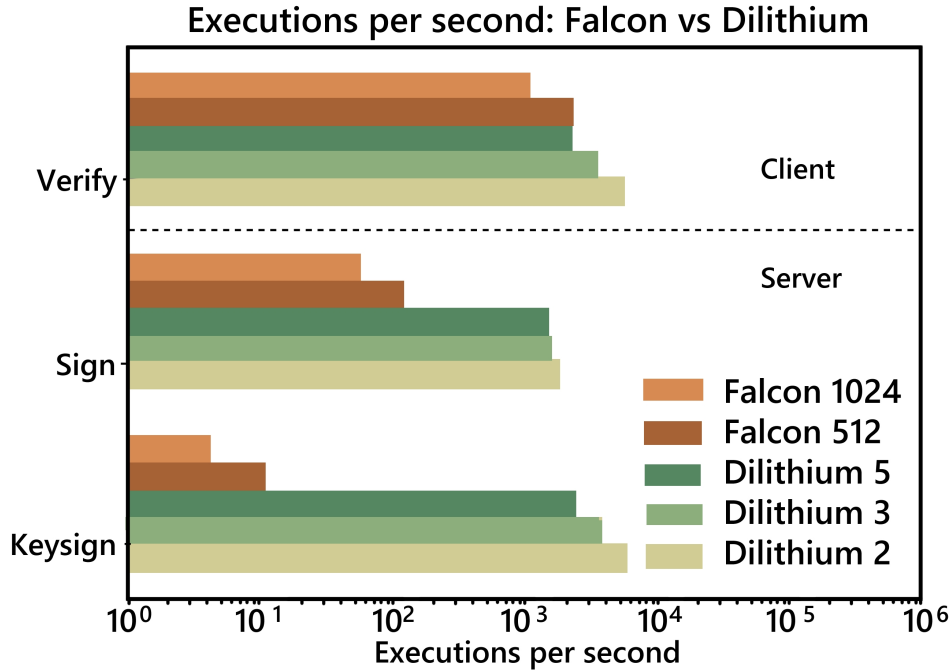


Figure 4.4: Comparison between the main operations of CRYSTALS-Dilithium and Falcon, including key generation, signing, and verification, across different security levels. CRYSTALS-Dilithium exhibits relatively balanced computational costs between operations, whereas Falcon shows significantly higher computational cost for key generation and signing, with verification remaining comparatively efficient. Reproduced from Figure 7 of [6], licensed under CC BY-NC-ND 4.0.

As illustrated in Figure 4.4, these differences in computational cost directly influence how post-quantum signature schemes behave in real-world deployments [6]. In high-performance environments such as data centers, ML-DSA is often attractive because of its comparatively balanced computational profile across signing and verification, although

its larger cryptographic artifacts increase communication overhead. In contrast, FN-DSA exhibits a more asymmetric computational profile in which key generation and signing are comparatively demanding, while verification is relatively efficient. Its deployment advantage therefore depends on authentication direction and endpoint role. In a conventional server-authenticated client-server setting, where the server signs and the client primarily verifies, FN-DSA can reduce the computational and communication burden on a constrained client at the cost of more demanding server-side signing. If the constrained endpoint itself must sign, however, this advantage may disappear or reverse [6, 17].

In a similar manner, ML-KEM, the NIST-standardized key encapsulation mechanism for post-quantum key exchange, achieves execution speeds comparable to classical approaches, but introduces increased communication overhead due to larger ciphertext sizes during the handshake process. Empirical observations further indicate that, although ML-DSA increases network load due to its larger payloads, it often achieves higher overall system throughput in high-capacity links by avoiding the computational bottlenecks that can limit FN-DSA-based deployments [6].

A further distinction arises between key encapsulation mechanisms and digital signature schemes in how they impact system behavior. Key encapsulation mechanisms such as ML-KEM are primarily used during the handshake phase and therefore affect connection establishment latency and communication overhead. In contrast, digital signature schemes are used during authentication and may also appear repeatedly in certificate-chain validation, software verification, or other trust-establishment workflows, making their computational and bandwidth characteristics more critical in sustained workloads. This distinction means that performance bottlenecks in post-quantum systems often arise from signature schemes rather than key encapsulation mechanisms.

This behavior is illustrated in Figure 4.5, which compares throughput for FN-DSA, ML-DSA, and ML-KEM across different parameter sets. The results show that throughput is not determined solely by artifact size. Although FN-DSA produces smaller signatures and reduces network load, its computational complexity can reduce effective throughput. By contrast, ML-DSA and ML-KEM may achieve higher throughput despite larger payloads, because their execution profiles are more efficient in the evaluated settings.

These effects become even more pronounced when considering large-scale systems. In high-speed networking environments, the computational overhead introduced by PQC can place substantial pressure on overall system capacity. Modern servers already dedicate a considerable portion of their computational resources to infrastructure-related tasks, with studies indicating that up to 30% of CPU cycles may be consumed by operations such as memory movement and packet processing [23, 24]. The additional burden introduced by post-quantum cryptographic operations can therefore push systems beyond their operational limits if not carefully managed.

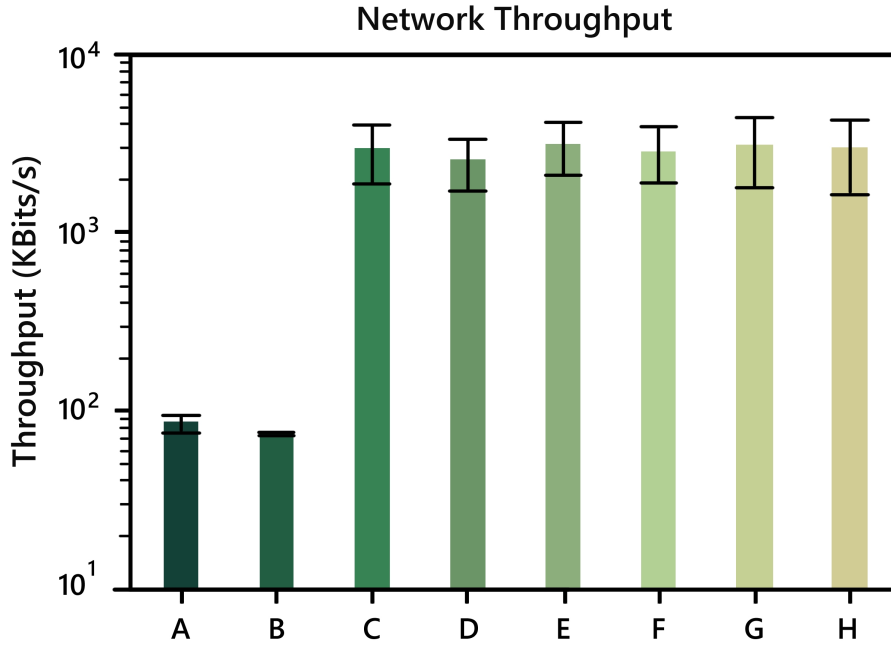


Figure 4.5: Network throughput of the pre-standardization Falcon, CRYSTALS-Dilithium, and CRYSTALS-Kyber implementations evaluated in [6] across different parameter sets. Falcon variants (A–B) achieve lower network load but significantly reduced throughput, whereas CRYSTALS-Dilithium (C–E) and CRYSTALS-Kyber (F–H) provide higher throughput despite increased communication overhead. Reproduced from Figure 8 of [6], licensed under CC BY-NC-ND 4.0.

This reflects a broader shift in system bottlenecks, where modern infrastructures are increasingly limited by computational capacity rather than raw network bandwidth. As a result, the efficiency of cryptographic computation becomes the dominant factor in determining system scalability.

In extreme cases, high-capacity communication links operating at 100 Gbit/s or 200 Gbit/s may experience substantial degradation when cryptographic processing is performed in software. This reduction in throughput reflects the cost of sustaining secure packet processing at line rate [7, 14].

In addition to computational cost, the increased size of post-quantum keys and signatures introduces additional challenges at the networking level. Larger payloads can lead to packet fragmentation, increased memory usage, and higher Input/Output (I/O) overhead, all of which reduce the efficiency of data transmission. These effects are particularly critical in latency-sensitive applications, where even small delays accumulate and affect overall system responsiveness. In practice, the combined impact of larger cryptographic material and increased processing complexity can shift cryptographic operations from the sub-millisecond to the millisecond regime, fundamentally altering system behavior [12]. As

a result, PQC affects not only security mechanisms, but also the performance characteristics of the underlying infrastructure.

Such system-level constraints are further influenced by the characteristics of the underlying hardware platform. To address these challenges, the effectiveness of post-quantum cryptographic deployments must be evaluated in the context of the systems in which they operate. The impact of a given algorithm depends heavily on factors such as hardware architecture, protocol design, and application requirements. At the hardware level, architectural optimizations can strongly affect execution efficiency. For instance, hardware-specific optimizations can significantly improve key-generation efficiency when combined with architecture-aware implementations, whereas such improvements may not be available or equally effective across different processor families [15]. Additionally, heterogeneous architectures combining Graphics Processing Units (GPUs) with specialized processing units highlight that large batch workloads can benefit substantially from hardware acceleration, while host CPUs may remain more efficient for low-volume requests because of accelerator overhead.

Recent independent measurements reinforce that these trade-offs are strongly platform- and operation-dependent. Evaluations of the standardized algorithms on constrained devices show that ML-KEM and ML-DSA can be practical even on embedded hardware, while their costs differ substantially across key generation, signing, verification, encapsulation, and decapsulation. In particular, ML-KEM exhibits comparatively predictable performance on embedded platforms, whereas ML-DSA signing is more demanding and SLH-DSA can impose substantially greater latency and energy cost on resource-constrained devices [91, 92]. These results reinforce that post-quantum algorithm selection should be based on the operations performed by a specific endpoint and the characteristics of the underlying hardware rather than on a single global performance ranking.

From a protocol perspective, the integration of PQC is also highly dependent on the layer at which security is applied. Mechanisms implemented at higher layers, such as TLS, are typically software-based and more difficult to offload, whereas link-layer protocols such as MACsec can leverage dedicated hardware engines to achieve line-rate encryption with minimal latency [14, 16].

Overall, algorithm selection in PQC cannot be made in isolation. It depends on the interaction between mathematical design, computational cost, communication overhead, implementation complexity, and deployment context. Since no single scheme is optimal across all environments, migration requires flexible strategies that can balance security guarantees with operational constraints. This motivates the hybrid and transitional approaches discussed next.

4.3 Hybrid Cryptography and the Transition to Post-Quantum Systems

The transition to PQC cannot be achieved through abrupt replacement of classical algorithms. Modern communication systems require interoperability across devices, protocols, libraries, and operators. Hybrid cryptography provides a practical transition strategy by combining classical and post-quantum mechanisms within the same protocol execution.

To manage these challenges, hybrid approaches have emerged as a practical strategy for enabling a gradual transition. This approach refers to the combination of multiple cryptographic primitives, typically including at least one classical algorithm and one post-quantum scheme, within a single protocol [8].

Hybrid Certificate	
Version	e.g. , v3
Serial Number	Unique ID assigned by CA
Signature Algorithm	e.g. , Hybrid (classical_ signature pq_signature)
Issuer	e.g. , Name of the CA
Validity	e.g. , Not Before / Not After Dates
Subject	e.g. , Name of Entity (Alice)
Hybrid Public Key Info - Classical Algorithm: e.g. , ECDSA (P-256) - Classical Public Key: e.g. , (P-256 key material) - Post-Quantum Algorithm: e.g. , ML-DSA-44 - Post-Quantum Public Key: e.g. , (ML-DSA-44 byte)	
Extensions (Optional) - Key Usage - Subject Alt Name (e.g. , DNS, email) - Constraints - Hybrid Indicator (e.g. , OID for p256_mldsa44)	
Signature - Classical Signature: e.g. , (ECDSA bytes) - Post-Quantum Signature: e.g. , (ML-DSA-44 bytes) - Combined: e.g. , Classical_signature pq_signature	

Figure 4.6: Hybrid X.509 certificate structure combining classical and post-quantum primitives. Adapted from Figure 3.8 of [1], with permission from the author.

Hybrid constructions are designed to diversify cryptographic assumptions. Their security depends not only on the component primitives, but also on the combiner, transcript binding, authentication, and protocol composition. With a robust combiner, a construction can be

designed to retain its intended security when at least one component remains secure [13, 93]. Operationally, hybrid mechanisms execute multiple key establishment or signature procedures during the same protocol interaction. This requires changes to handshake messages, key scheduling, transcript binding, and validation logic, since post-quantum KEMs do not follow the same interaction model as classical DH-based exchanges.

Hybridization can also be applied to certificate-based authentication. In a hybrid X.509 certificate, classical and post-quantum public-key and signature material can coexist within the certificate structure, allowing existing trust infrastructures to support a gradual transition toward post-quantum authentication. Figure 4.6 illustrates this approach.

Such hybrid certificates provide a transitional mechanism in which classical and post-quantum authentication can coexist while certificate authorities, validation software, and endpoints are progressively updated. The additional cryptographic material can, however, increase certificate size and validation complexity.

For key establishment, a protocol handshake may involve both a classical key exchange, such as ECDH, and a post-quantum key encapsulation mechanism such as ML-KEM. In more advanced configurations, hybrid deployments such as triple-hybrid IPsec tunnels can use protocols like IKEv2 to negotiate shared secrets derived from classical, post-quantum, and QKD-based sources simultaneously [13]. Figure 4.7 illustrates how multiple key establishment mechanisms can operate in parallel within such hybrid constructions.

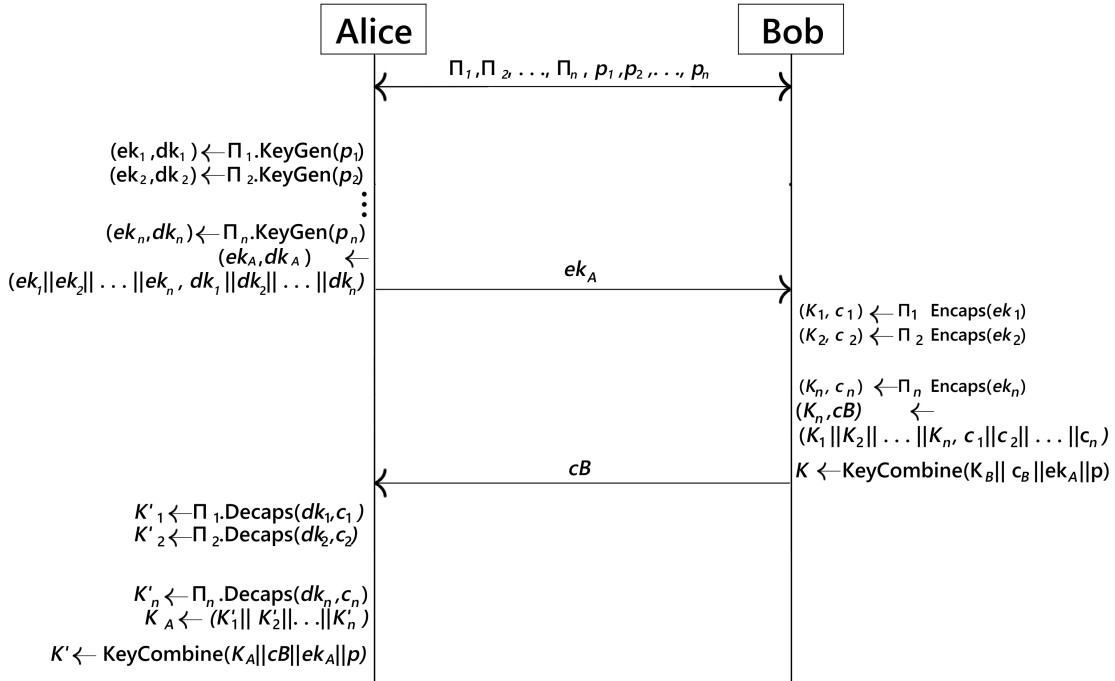


Figure 4.7: Parallel hybrid key establishment using classical, post-quantum, and optional quantum-based mechanisms. Adapted from Figure 3.9 of [1], with permission from the author.

As shown in Figure 4.7, different key establishment mechanisms are executed concurrently, each producing an independent shared secret. These may include classical key exchange, post-quantum key encapsulation, and potentially QKD. The final session key is then derived by combining all individual secrets, ensuring that the resulting construction remains secure as long as at least one of the underlying mechanisms remains uncompromised.

The resulting shared secrets are combined and processed through a KDF, typically based on constructions such as HKDF, to produce the final session keys used for secure communication [53]. In hybrid settings, the outputs of multiple key establishment mechanisms may be concatenated or combined through cascade-based approaches before key derivation. In cascade constructions, the key material is updated iteratively across multiple rounds until a final shared secret is obtained. This ensures that the resulting communication session remains secure as long as at least one of the underlying mechanisms remains uncompromised [13].

A practical example of this approach appears in IPsec, where post-quantum material can be integrated through pre-shared post-quantum keys. In this model, the post-quantum key exchange is performed outside the real-time IKEv2 handshake and the resulting post-quantum pre-shared key is stored by both peers. During tunnel establishment, the classical key exchange proceeds within IKEv2, and the post-quantum pre-shared material is subsequently mixed with the classical key material during key derivation. This preserves compatibility with existing IPsec workflows and reduces the direct impact of post-quantum operations on handshake latency, while requiring additional infrastructure for the exchange, storage, and management of post-quantum pre-shared keys [9].

Table 4.1: AEAD schemes involved in the quantum migration. Adapted from Table 3.4 of [1], with permission from the author.

AEAD Cipher	Usage	Classical (λ)	Post-Quantum (λ)
AES-GCM-128	TLS, IPsec, MACsec	128	64
AES-GCM-256	TLS, IPsec, MACsec	256	128
AES-CCM-128	TLS	128	64
AES-CCM-256	TLS	256	128
ChaCha20–Poly1305	TLS	256	128
ASCON-128a	Lightweight	128	64

Once these keys are established, the resulting session keys are used to protect application data through symmetric encryption mechanisms, typically using primitives such as AES-256-GCM. In high-throughput deployments, this data protection can be hardware accelerated, as examined in Chapters 5 and 6 [8]. Table 4.1 summarizes the AEAD schemes involved in the quantum migration, highlighting the symmetric encryption mechanisms that continue to support high-throughput data protection once secure session keys have been established.

As shown in Table 4.1, the transition to PQC does not fundamentally alter the role of symmetric authenticated encryption. Instead, AEAD schemes such as AES-GCM, AES-Counter with CBC-MAC (CCM), ChaCha20-Poly1305, and lightweight alternatives like Authenticated Security Combined with Nonce (ASCON) continue to provide efficient confidentiality and integrity protection in the data plane.

In this context, the principal generic quantum-search concern for symmetric cryptography can be mitigated by selecting sufficiently large key sizes, typically favoring 256-bit keys where long-term security margins are required. As discussed in Chapter 3, the corresponding Grover-based reductions represent idealized query-complexity estimates rather than direct practical attack costs. This reflects an important distinction in the migration process. While quantum-vulnerable asymmetric primitives require replacement, symmetric authenticated encryption remains structurally usable with appropriately selected security parameters.

In addition to classical and post-quantum combinations, such constructions can incorporate additional sources of key material, such as QKD. In those settings, multiple independent mechanisms contribute to the final shared secret, creating a layered security model. By integrating classical key exchange, PQC, and QKD-derived keys, these approaches reduce reliance on any single cryptographic assumption and improve overall system resilience.

QKD can also contribute key material in specialized hybrid deployments. Unlike PQC, which relies on computational hardness assumptions, QKD derives its security from quantum-mechanical properties of the communication channel. This guarantee applies specifically to the key-distribution process and depends on implementation assumptions, device behavior, and the availability of an authenticated classical channel. QKD therefore does not replace PQC, since it does not provide general-purpose digital signatures, scalable authentication, or broad Internet-wide deployment. It is best understood as a complementary mechanism for specialized links where the required physical and operational infrastructure is available [13, 39].

Hybrid cryptography also introduces costs. Executing multiple primitives during a protocol exchange increases computation, message size, and implementation complexity. Larger keys, ciphertexts, and signatures can increase bandwidth consumption and may lead to packet fragmentation, while additional cryptographic operations can increase handshake latency. These effects are particularly important in high-throughput systems, where even small per-connection overheads accumulate at scale [7, 9].

Experimental evaluations indicate that this overhead does not necessarily make hybrid migration impractical. Hybrid signature combinations, such as ECDSA with ML-DSA or FN-DSA, can provide competitive signing and verification throughput when the selected algorithms and deployment architecture match the target environment [9].

Beyond performance considerations, these systems introduce significant integration complexity. Integrating classical, post-quantum, and quantum-based mechanisms requires

coordination across different software layers, cryptographic libraries, and infrastructure components, including certificate authorities and key management systems. Ensuring interoperability across heterogeneous environments becomes more challenging as more primitives are deployed. At the same time, certain technologies such as QKD impose additional constraints related to deployment scale and physical infrastructure, which further complicates system design. In particular, QKD systems are affected by physical limitations such as fiber attenuation and a secret key rate that decreases with transmission distance, restricting their practical deployment range [13, 39]. These factors show that hybrid cryptography is not only a security solution, but also a significant engineering challenge.

Despite these challenges, many current hybrid deployments are transitional risk-reduction mechanisms, although hybrid designs may persist where defense in depth, interoperability, or multiple independent key sources remain operationally useful [31, 93]. By maintaining compatibility with existing systems and providing fallback security guarantees, hybrid constructions allow organizations to adopt new cryptographic technologies incrementally. As confidence in post-quantum schemes increases and optimized implementations become more widely available, systems are expected to move toward fully quantum-resilient architectures with reduced reliance on hybrid combinations [13].

Collectively, hybrid cryptography plays a key role in the evolution of secure communication systems. It enables classical and post-quantum mechanisms to operate together, providing a practical path for managing the transition. At the same time, it shows that moving toward quantum-resistant cryptography is not simply a matter of replacing algorithms, but requires a broader system-level transformation that must balance security, performance, and operational constraints. These challenges motivate the migration strategies examined in the following section.

4.4 Migrating to Post-Quantum Cryptography

The transition to PQC represents more than a simple upgrade of cryptographic algorithms. It is a system-wide transformation that affects the entire digital security ecosystem. In this context, migration refers to the gradual replacement of classical asymmetric cryptographic primitives, such as RSA, DH, and ECC, with quantum-resistant alternatives based on different mathematical assumptions. While symmetric cryptography can be adapted by increasing key sizes, asymmetric systems require a deeper redesign because their underlying security mechanisms must be replaced entirely. Accordingly, migration involves not only the substitution of algorithms, but also the reconfiguration of protocols, key management systems, and the broader PKI that supports modern communication.

This transformation is inherently complex because cryptographic mechanisms are tightly integrated into existing systems. Modern infrastructures rely on long-established standards and protocols that require interoperability and backward compatibility across diverse envi-

ronments. Introducing new primitives into such ecosystems cannot be performed abruptly, as it may break communication between systems that support different mechanisms. Simultaneously, post-quantum schemes impose additional computational and communication overhead, which complicates their integration into performance-sensitive applications. For these reasons, migration is a non-trivial process that must be carefully staged and managed over time.

For the purposes of this book, migration can be viewed as a multi-stage process that evolves alongside technological and standardization developments. The initial phase involves the deployment of hybrid cryptographic mechanisms, where classical and post-quantum primitives coexist within the same protocol to ensure backward compatibility and maintain security guarantees [13]. This phase is followed by an integration and validation stage. During this stage, existing protocol suites, such as TLS, IPsec, and MACsec, are adapted to support quantum-resistant primitives and evaluated in realistic deployment environments [13, 16]. The final stage corresponds to the emergence of fully crypto-agile systems, in which post-quantum algorithms are deployed as primary mechanisms. In this setting, hybrid constructions may eventually be replaced once sufficient confidence in their security and performance has been established.

This transition is reflected not only at the architectural level, but also in the key-establishment mechanisms that must coexist during migration. Table 4.2 presents representative classical key-exchange mechanisms, post-quantum key encapsulation mechanisms, and hybrid combinations involved in this process. These hybrid approaches combine established schemes such as ECDH with post-quantum key encapsulation mechanisms like ML-KEM, enabling the construction of dual- or multi-layered security models that rely on independent cryptographic assumptions.

Table 4.2: Key exchange mechanisms involved in the quantum migration. Adapted from Table 3.2 of [1], with permission from the author.

KEX / KEM	PK (B)	SK (B)	Encap (B)	Classical (λ)	Post-Quantum (λ)	NIST Level
ECDH P-256	65	32	65	128	0	–
ECDH P-384	97	48	97	192	0	–
ECDH P-521	133	66	133	256	0	–
MODP 2048 DH	256	256	256	112	0	–
X25519	32	32	32	128	0	–
ML-KEM-512	800	1632	768	128	128	L1
ML-KEM-768	1184	2400	1088	192	192	L3
ML-KEM-1024	1568	3168	1568	256	256	L5
Hybrid P-256 + ML-KEM-512	865	1664	833	128	128	L1
Hybrid P-384 + ML-KEM-768	1281	2448	1185	192	192	L3
Hybrid P-521 + ML-KEM-1024	1701	3234	1701	256	256	L5

The security-strength values in Table 4.2 are intended as comparative parameter-level reference points rather than complete security bounds for the resulting protocols. In hybrid constructions, overall security also depends on how the component mechanisms are combined, authenticated, and incorporated into the protocol key schedule.

Recent production deployments show that this transition has moved beyond isolated experimental implementations. Hybrid X25519MLKEM768 key establishment is now being introduced across several large operational ecosystems. Google Cloud supports it on major API endpoints and load balancers, Apple enables it by default in current TLS networking APIs, AWS has deployed ML-KEM-based hybrid TLS to security-sensitive service endpoints, and Akamai has deployed hybrid post-quantum protection across client-to-edge, inter-edge, and edge-to-origin communication paths [94, 95, 96, 97]. These deployments indicate that the practical migration challenge is increasingly shifting from algorithm availability toward interoperability, cryptographic agility, software and protocol upgrades, and coordinated deployment across heterogeneous infrastructure.

Migration is increasingly shaped not only by cryptographic standardization, but also by security-authority guidance that translates the quantum threat into concrete lifecycle, procurement, and system requirements. These frameworks do not all have the same regulatory scope. In the United States, Commercial National Security Algorithm Suite (CNSA) 2.0 defines quantum-resistant requirements primarily for U.S. National Security Systems rather than a general mandate for commercial systems. CNSA 2.0 specifies AES-256 for symmetric protection, ML-KEM-1024 for key establishment, ML-DSA-87 for general-purpose digital signatures, and permits LMS and XMSS in specific software- and firmware-signing applications. Its current transition schedule requires new National Security System acquisitions to support CNSA 2.0 from 2027, phases out equipment and services unable to support the suite by the end of 2030, and mandates CNSA 2.0 algorithms by the end of 2031, within the broader objective of making National Security Systems quantum-resistant by 2035 [32]. Importantly from a system-design perspective, NSA also identifies cryptographic agility as necessary for this transition and recommends that equipment retain sufficient memory, processing capability, and protocol capacity to accommodate future cryptographic changes through software updates [32].

European migration guidance follows a more distributed but similarly staged approach. The 2025 Coordinated Implementation Roadmap for the Transition to Post-Quantum Cryptography calls on all EU Member States to start transitioning by the end of 2026 and identifies the protection of critical infrastructure as a priority to be transitioned as soon as possible and no later than the end of 2030 [67]. A joint statement from cybersecurity authorities in 18 EU Member States similarly recommends cryptographic inventory, risk-oriented prioritization based on information sensitivity and protection lifetime, explicit consideration of harvest-now-decrypt-later threats, migration planning and budgeting, crypto-agility, and hybrid deployment during the transition [31]. At the technical level, the applicable European Cybersecurity Certification Group Agreed Cryptographic Mechanisms

guidance recommends prioritizing post-quantum protection for long-lived confidentiality and deploying post-quantum asymmetric mechanisms in hybrid mode with established classical mechanisms, while also adapting symmetric key and hash lengths to the quantum threat [98]. Current Federal Office for Information Security (BSI) guidance for TLS and IKEv2 likewise anticipates the introduction of quantum-safe mechanisms through hybrid use with recommended classical mechanisms as suitable protocol standards mature [99, 100].

Recent French and European guidance further illustrates how this transition is moving from algorithm selection toward protocol and system engineering. French National Cybersecurity Agency (ANSSI) recommends that organizations begin quantum-risk analysis and cryptographic inventory now, prioritize critical cryptographic use cases, and incorporate post-quantum support into normal system-renewal and procurement cycles [101]. Its 2026 guidance on crypto-agility is explicitly directed at developers and system architects and emphasizes the need for systems to accommodate future changes in cryptographic mechanisms without major redesign [33]. ANSSI has also published dedicated 2026 transition guidance for TLS 1.3, IPsec, and Secure Shell (SSH), reflecting the increasing maturity of protocol-level migration work [101]. In parallel, European Union Agency for Cybersecurity (ENISA)’s 2026 hybridization report maps ongoing standardization across TLS, IKEv2, SSH, Cryptographic Message Syntax (CMS)/Secure/Multipurpose Internet Mail Extensions (S/MIME), PKI, and related mechanisms; the report is a standards-status analysis rather than a normative ENISA or ECCG recommendation [93]. Taken together, these developments reinforce the system-level view adopted in this book. Successful post-quantum migration requires coordinated planning across cryptographic inventory, protocols, software, hardware capacity, trust infrastructure, procurement, and system lifecycles rather than the isolated replacement of individual cryptographic primitives.

Against this policy and deployment background, a key requirement throughout this process is crypto-agility, which refers to the ability of a system to adapt to new cryptographic primitives without requiring major changes to its architecture [13]. In a rapidly evolving threat landscape, where both classical and quantum attacks must be considered, this flexibility enables systems to respond to newly discovered vulnerabilities or advances in cryptanalysis. This aspect is particularly important in the post-quantum context, where standardized algorithms may continue to evolve, and where the ability to switch or combine primitives dynamically provides an important layer of resilience. Crypto-agile systems also enable the deployment of hybrid and triple-hybrid constructions that can be adapted at runtime based on application requirements or emerging threats [13].

Despite this structured progression, migration introduces a range of technical challenges that affect both performance and system design. Post-quantum algorithms typically require additional computation and introduce a larger spatial footprint through larger public keys, ciphertexts, and signatures. For example, ML-DSA public keys are approximately twenty times larger than those for ECDSA, which can contribute to packet fragmentation

and increased bandwidth consumption [6, 9]. These costs must be accommodated by infrastructures that already operate under significant processing and communication constraints. Their system-level implications are examined in Chapter 5 and evaluated experimentally in Chapter 6.

Beyond performance considerations, the migration process also introduces a number of security and operational risks. The HNDL threat discussed in Section 3.4 reinforces the need for early migration, while operational inertia can prolong reliance on outdated and quantum-vulnerable algorithms [13]. Further risks arise from performance bottlenecks. Poorly optimized post-quantum implementations can introduce latency and increase computational load, potentially affecting performance-critical applications such as large-scale data processing and AI workloads [12, 21]. Furthermore, newly standardized algorithms may still be exposed to implementation-level vulnerabilities, including side-channel attacks or design oversights that appear during early deployment phases [87, 88, 89].

These migration challenges also have architectural consequences, since the additional computational and communication requirements of post-quantum mechanisms affect where cryptographic functions can be executed efficiently. The resulting infrastructure-level adaptations, including hardware-assisted execution and the distribution of cryptographic processing across system components, are examined in Chapter 5 and evaluated through applied deployments in Chapter 6.

Beyond hardware-based optimizations, migration strategies also rely on hybrid and multi-protocol constructions that provide redundancy and flexibility. By combining classical, post-quantum, and in some cases quantum-based key establishment mechanisms, these methods ensure that communication remains secure as long as at least one of the underlying primitives remains uncompromised. Furthermore, the integration of post-quantum cryptographic libraries into virtualized environments enables cloud-native systems to adopt new security mechanisms without sacrificing scalability. Such approaches highlight the importance of designing systems that remain adaptable to both current and future cryptographic requirements [18, 19].

The distinction between short-term and long-term migration mechanisms is especially visible in IKEv2. Pre-shared post-quantum key approaches, such as those based on RFC 8784, provide a transitional mechanism by mixing externally provisioned key material into the protocol. However, this approach offers limited scalability and weaker support for dynamic rekeying, since the post-quantum material must be distributed and managed outside the normal key-exchange process. By contrast, RFC 9370 integrates multiple successive key exchanges directly into IKEv2, enabling dynamic negotiation, rekeying, multiple hybrid combinations, and stronger crypto-agility. This makes RFC 9370 a more suitable long-term mechanism for quantum-resilient IPsec migration [13, 102, 103].

Looking ahead, the long-term objective of migration is the realization of fully integrated, high-performance, and quantum-resilient infrastructures. These environments combine

crypto-agile design with optimized hardware support, enabling secure communication across the entire edge-to-cloud continuum. In this context, communication across the infrastructure is protected through multilayer encryption strategies, for example by combining link-layer mechanisms such as MACsec with network-layer protocols such as IPsec, ensuring end-to-end protection across different abstraction layers [16].

More broadly, such infrastructures are expected to rely on a hybrid security landscape in which multiple cryptographic paradigms coexist. In particular, PQC provides scalable authentication and key establishment, while QKD can contribute key-distribution material with information-theoretic security guarantees under the assumptions of the QKD protocol, including an authenticated classical channel and appropriate device and implementation security. QKD therefore complements rather than replaces post-quantum authentication and digital signatures. In specialized links where the required quantum and key-management infrastructure is available, combining these mechanisms can strengthen long-term confidentiality while preserving scalable authentication [13, 39]. This perspective reflects a broader shift toward infrastructures in which security is not treated as an isolated component, but as a core and continuously evolving part of the system as a whole.

In summary, managing the transition to PQC requires a careful balance between security, performance, and operational constraints. Rather than a simple replacement of algorithms, it involves coordinated changes to protocols, infrastructure, and long-term system design. This complexity highlights that the development of quantum-resilient systems is fundamentally a system-level challenge, requiring both technical innovation and strategic decision-making. These considerations naturally lead to the broader implications of post-quantum migration, which are examined in the following section.

4.5 What Post-Quantum Migration Means for Systems

The transition to PQC changes not only which algorithms are used, but also how cryptographic work is integrated into protocols and infrastructures. Classical secure communication systems were designed around compact public-key operations for authentication and key establishment, followed by efficient symmetric encryption for bulk data protection. Post-quantum mechanisms preserve this functional separation, but they alter its cost profile by introducing larger keys, ciphertexts, and signatures, as well as different computational requirements.

At the protocol level, these changes primarily affect operations such as handshakes, certificate validation, key establishment, algorithm negotiation, and rekeying. These procedures must be adapted to support post-quantum or hybrid mechanisms while preserving interoperability with existing deployments. Once session keys are established, high-throughput data protection can continue to rely on symmetric authenticated encryption, provided that

appropriate key sizes are selected. This distinction explains why migration must address both the cryptographic primitives themselves and the protocols that coordinate their use.

At the system level, larger cryptographic artifacts and more complex operations can affect bandwidth, memory movement, packet fragmentation, latency, and resource utilization. These effects are not uniform across deployment environments. A mechanism that is practical in a data-center environment may be unsuitable for a constrained endpoint, while a compact signature may reduce communication overhead but increase signing complexity. Algorithm selection must therefore be evaluated together with protocol behavior, endpoint capability, and operational constraints [6, 9].

These constraints also influence where cryptographic operations are executed. In some environments, post-quantum processing can remain in software, especially for moderate connection rates or endpoints with sufficient processing capacity. In high-throughput infrastructures, however, the additional cost of post-quantum handshakes, certificate validation, and key-management operations may motivate closer integration with hardware-assisted networking and security components. The detailed architectural consequences of this shift, including offloading, programmable network interfaces, and infrastructure-level acceleration, are examined in the following chapters [12, 104].

Post-quantum migration also requires crypto-agility. Since standards, implementations, and cryptanalytic confidence will continue to evolve, systems must support algorithm negotiation, parameter updates, certificate replacement, and controlled deprecation of vulnerable mechanisms. Hybrid deployments provide one transitional method for maintaining compatibility and reducing reliance on a single assumption, but long-term deployment requires architectures that can evolve without complete redesign.

Post-quantum migration is therefore a system-level process. It begins with new mathematical foundations, but its success depends on protocol integration, operational planning, implementation security, and deployment flexibility. The next chapter turns from migration strategy to the infrastructure in which these mechanisms must operate.

Chapter 5

From Infrastructure to Application in Post-Quantum Cryptography

This chapter examines how PQC is realized within modern digital infrastructures. Building on the algorithmic perspective developed in Chapter 4, the focus shifts toward the system-level context in which these mechanisms must operate. It first considers cryptography as a foundational component of contemporary infrastructure and then analyzes how architectural and protocol-level constraints shape post-quantum deployment. The discussion then introduces the distinction between infrastructure-level and applied PQC, clarifying how mathematical mechanisms are mapped onto protocols, software paths, and hardware execution models. Finally, the chapter illustrates this mapping through representative protocol and system mechanisms, establishing the infrastructure baseline for the applied evaluations that follow.

Source note. Parts of the infrastructure discussion and selected illustrations in this chapter draw on Chapters 4 and 6 of [1] and Chapter 3 of [2]. The material has been reorganized and extended with the additional sources cited throughout.

5.1 Cryptography in Modern Infrastructure

In modern digital systems, cryptography is no longer an isolated mathematical primitive. It operates across communication protocols such as TLS, IPsec, and MACsec, trust frameworks such as PKI, operating-system networking stacks, and increasingly specialized hardware components such as NICs, SmartNICs, and DPUs. In this environment, cryptography functions as a core architectural element that spans software, protocols, and hardware [50, 52, 62].

This level of integration has direct consequences for the transition to PQC. Since public-key mechanisms are embedded in authentication, key establishment, certificates, and

protocol negotiation, replacing quantum-vulnerable primitives requires coordinated changes across the system stack. The transition therefore cannot be treated as a local algorithm substitution. It affects protocol formats, key-management logic, software libraries, hardware execution paths, and operational workflows.

When PQC is considered in real-world infrastructure, additional constraints become visible. Post-quantum schemes introduce larger public keys, ciphertexts, and signatures, as well as different computational profiles. These properties affect bandwidth, memory movement, packet processing, latency, and scalability. As a result, the practical deployment of PQC depends not only on mathematical security, but also on whether the surrounding infrastructure can absorb the additional computational and communication costs [9, 36, 37].

5.2 Infrastructure as a Constraint on Post-Quantum Cryptography

In modern computing environments, cryptography operates within tightly defined system and protocol boundaries. These boundaries directly constrain how post-quantum schemes can be deployed. Cryptographic processing is embedded in operating-system networking stacks, protocol implementations, certificate validation logic, and hardware execution paths. As a result, PQC cannot be introduced only by replacing one algorithm with another. It must fit within the data structures, packet formats, negotiation mechanisms, and performance constraints of existing infrastructure.

This execution context is illustrated in Figure 5.1, which shows how packets traverse the Linux networking stack during ingress and egress. The figure highlights that packets pass through a structured pipeline involving the network interface, kernel networking components, routing logic, transport processing, and application-facing interfaces. Cryptographic mechanisms must integrate into this pipeline without disrupting throughput, compatibility, or policy enforcement [2, 105].

Figure 5.1 highlights how the protocol-level separation introduced in Chapter 2 maps onto a concrete system execution path. Session-establishment functions such as authentication, policy evaluation, and key negotiation are handled through control-plane logic, while the resulting symmetric protection is applied to continuous traffic in the data path. In optimized deployments, selected high-volume operations may be transferred to hardware-assisted paths to reduce repeated involvement of the host CPU.

In high-performance environments, this packet-processing path may be further optimized by bypassing the traditional kernel networking stack. Frameworks such as the Data Plane Development Kit (DPDK) enable packet processing directly in user space, avoiding interrupt-driven kernel processing and reducing overhead associated with socket buffers and

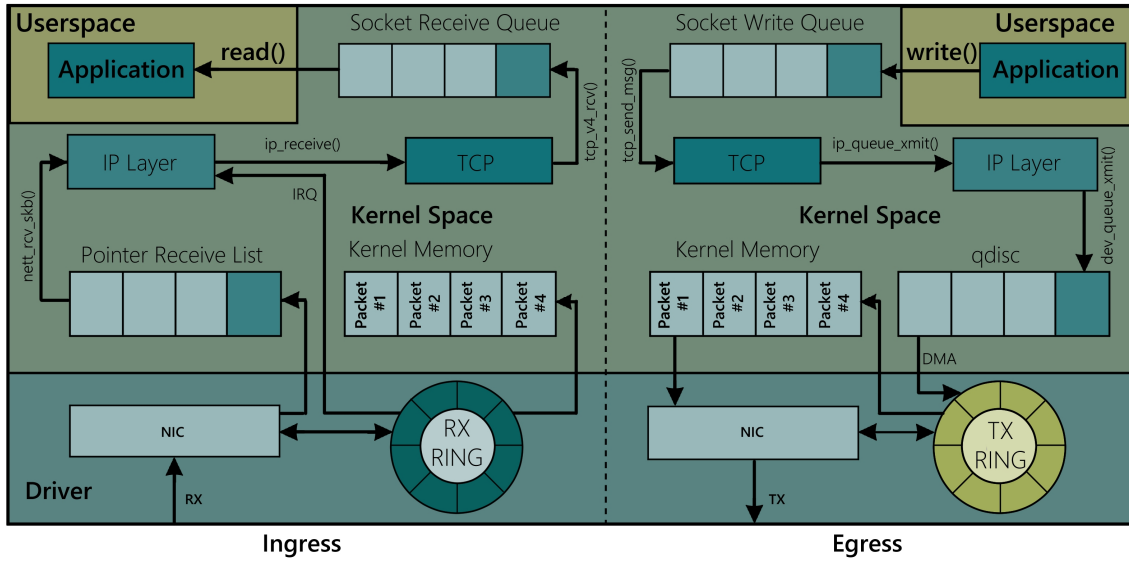


Figure 5.1: The path of a packet through the Linux kernel during ingress and egress. Adapted from Figure 3.3 of [2], with permission from the author.

context switches. Instead of traversing multiple kernel layers, packets can be received from the network interface and processed directly by user-space applications through polling-based mechanisms. This approach enables significantly higher packet throughput and improved scalability across multiple cores, which is essential for sustaining cryptographic processing at line-rate speeds in data-center environments. However, bypassing the kernel also reduces compatibility with conventional debugging and monitoring tools, introducing additional operational complexity [2].

A further performance consideration arises from memory management. High-speed packet-processing frameworks often rely on large memory pages, or HugePages, rather than standard memory pages used by the operating system. By allocating larger contiguous memory regions, these approaches reduce address-translation overhead and improve cache efficiency, which is particularly important for workloads involving frequent memory access. In the context of PQC, where larger keys, signatures, and ciphertexts increase memory movement and I/O demands, such optimizations can help maintain overall system performance [2].

Infrastructure constraints also depend on traffic patterns. North-south traffic refers to communication between external clients and internal data-center resources, typically traversing perimeter components such as firewalls, load balancers, and intrusion-detection systems. East-west traffic occurs within the data center between servers, services, containers, or virtual machines. In modern cloud and hyperscale environments, east-west traffic often dominates total bandwidth consumption due to distributed storage systems, microservice architectures, and parallel computing workloads that require frequent inter-node communi-

cation [106, 107]. This internal communication must be protected at high throughput and low latency, which places additional pressure on cryptographic processing pipelines.

These constraints are reinforced by performance limitations that dominate practical deployment. Modern systems already operate under significant resource pressure, with a substantial portion of computational capacity consumed by infrastructure-related tasks such as memory movement and packet processing. This pre-existing overhead is often described as the “data center tax” and has been reported to account for roughly 30% of CPU cycles in large-scale systems [23]. Post-quantum cryptographic schemes introduce additional computational complexity, especially in lattice-based constructions that rely on polynomial arithmetic, matrix-vector multiplication, sampling, and hashing. They also increase the spatial footprint of cryptographic material, which raises bandwidth usage, I/O overhead, and latency [9].

At the same time, a growing gap has emerged between network bandwidth and general-purpose processing capability. Modern network interfaces operate at rates that make host-based packet processing increasingly difficult to sustain at line rate. As a result, the bottleneck in secure communication shifts from link capacity to computational throughput. This shift is especially important for PQC, whose additional processing and communication costs amplify the limitations of traditional CPU-centric designs [24, 104].

In high-throughput environments, these pressures motivate a shift from CPU-centric toward network-centric execution, in which selected networking and security functions are moved closer to the data path while general-purpose processors remain available for application workloads. At high data rates, this protocol separation becomes an execution-placement problem. Post-quantum control-plane functions can remain in programmable software paths, while continuous symmetric traffic protection may be mapped to hardware-assisted execution on components such as DPUs and SmartNICs [7, 12].

DPUs and SmartNICs extend conventional network interfaces by integrating embedded processors, programmable packet-processing pipelines, and dedicated acceleration engines. Unlike traditional NICs, which primarily handle physical and link-layer functions, these devices can execute networking, security, and storage-related operations closer to the data path [1, 104].

Figure 5.2 illustrates this architectural shift by comparing a conventional NIC-based server with a SmartNIC-based design. In the conventional model, packet processing and security operations remain primarily attached to the host system. In the SmartNIC-based model, programmable processing capabilities are moved closer to the network interface, allowing selected networking and security functions to be executed nearer to the data path.

In the context of PQC, this architecture helps preserve host resources while allowing high-throughput data protection to continue once quantum-resilient control-plane mechanisms have established the required session keys.

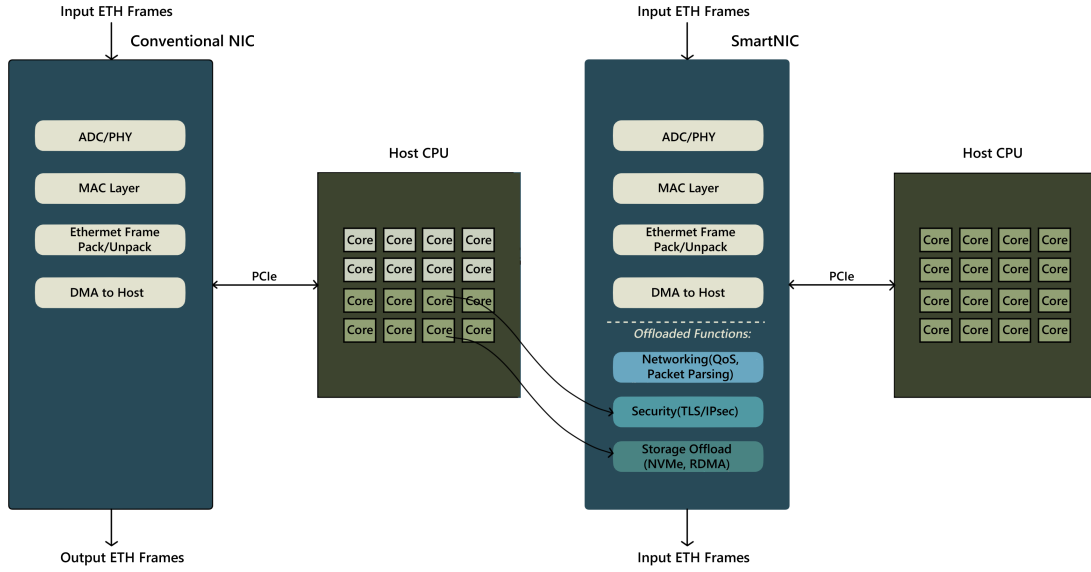


Figure 5.2: Schematic comparison of a conventional NIC-based server and a SmartNIC-based server architecture. Adapted from Figure 4.1 of [1], with permission from the author.

At large scale, these constraints directly determine the feasibility of PQC. Data-center, cloud, and AI environments must support large numbers of concurrent secure flows while transferring substantial volumes of data between storage, compute, and service nodes [17]. In such settings, even modest per-operation overheads introduced by post-quantum mechanisms can accumulate into system-wide bottlenecks. Larger keys and signatures increase pressure on memory and I/O systems, while frequent authentication and certificate validation can increase the demand for signature generation and verification [21]. Consequently, a cryptographic scheme that appears efficient in isolation may become impractical unless the surrounding infrastructure can absorb its cumulative costs.

5.3 Infrastructure-Level and Applied Post-Quantum Cryptography

For the purposes of this discussion, a distinction can be drawn between *infrastructure-level PQC* and *applied PQC*. This distinction is not intended as standardized terminology. It is used here as a conceptual tool to separate two related questions concerning where post-quantum mechanisms must be integrated, and how they behave when deployed in real systems.

Infrastructure-level PQC refers to the integration of post-quantum mechanisms into the architectural foundations of digital systems. This includes communication protocols, certificate ecosystems, key-management frameworks, operating-system networking stacks, and hardware-assisted execution paths. At this level, the main concern is compatibility

with the structures that already coordinate trust, authentication, key establishment, and data protection.

Applied PQC refers to the behavior of these mechanisms under operational constraints. This includes their effect on latency, throughput, memory movement, I/O overhead, packet fragmentation, hardware acceleration, virtualization, and scalability. A scheme that is mathematically secure may still be difficult to deploy if its computational or communication costs exceed the capabilities of the target environment.

This distinction is useful because mathematical security alone does not determine deployability. Theoretical PQC focuses on constructing primitives from problem families believed to resist both classical and quantum adversaries. Infrastructure-level PQC asks how those primitives can be integrated into existing systems. Applied PQC asks whether the resulting deployment can operate efficiently, securely, and reliably under real workload conditions.

In practice, these levels interact because protocol structures must be mapped onto concrete software, networking, and hardware execution paths. Successful deployment therefore depends on both cryptographic design and system architecture.

This perspective changes how PQC is understood. It is not only an algorithmic transition, but also an infrastructure problem. Achieving quantum-resilient security requires coordinated changes across protocols, software stacks, hardware accelerators, key-management systems, and operational processes. This framing prepares the ground for the applied examples discussed in the following section.

5.4 What Applied Post-Quantum Cryptography Means in Practice

The distinction outlined above becomes concrete at the protocol level, where post-quantum mechanisms must be incorporated into existing key-management procedures and communication architectures.

In protocols such as IPsec, post-quantum integration focuses on IKEv2. Hybrid approaches combine classical mechanisms, such as DH, with post-quantum schemes such as ML-KEM to derive shared secrets. Similarly, in MACsec, integration is achieved through the MKA, where hybrid master keys can be established through software-defined control planes or extended handshake mechanisms. This key-management logic is illustrated in Figure 5.3, which shows how the MKA protocol derives the cryptographic material used to protect Ethernet traffic. The figure highlights both pre-shared and dynamically established keying modes, where either a pre-shared Connectivity Association Key (CAK) and Connectivity Association Key Name (CKN), or a dynamically derived Master Session Key (MSK), serve as the initial input. Through a sequence of key derivation steps, this material is transformed into the SAK, along with auxiliary keys such as the KEK and ICK. The SAK is then used

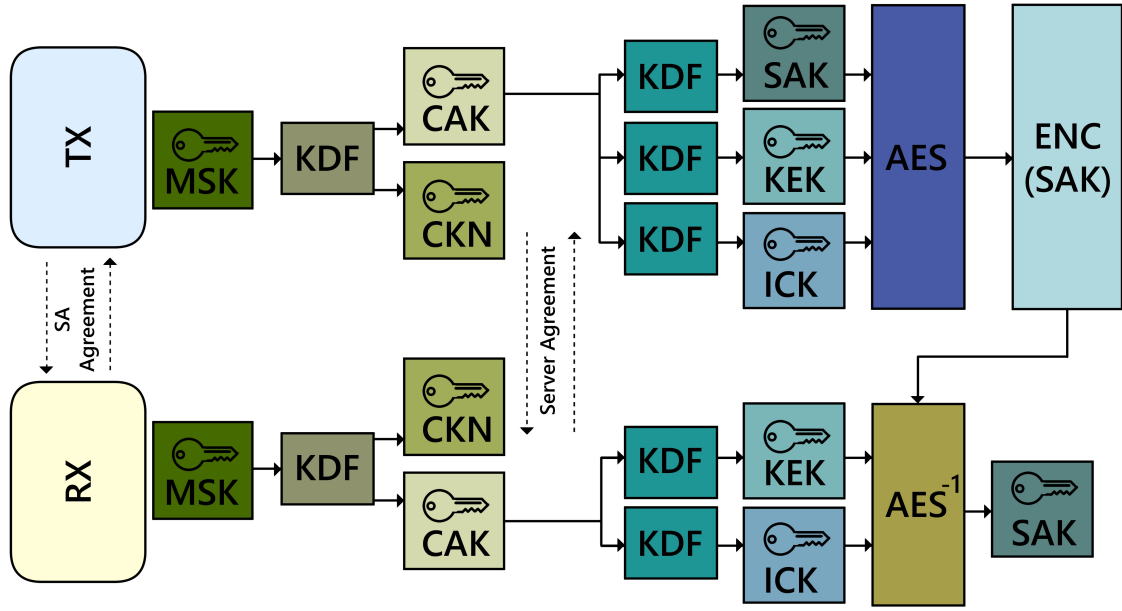


Figure 5.3: MKA hierarchy illustrating both pre-shared and dynamic key derivation modes, and the generation of Secure Association Key (SAK), Key Encryption Key (KEK), and Integrity Check Key (ICK) used for data-plane protection. Adapted from Figure 6.3 of [1], with permission from the author.

to protect data-plane traffic through symmetric encryption, illustrating the separation between control-plane key establishment and data-plane protection [1, 64].

This hierarchy also illustrates the practical role of key derivation in applied PQC. Shared secrets established through classical, post-quantum, or hybrid mechanisms are not used directly as data-plane keys. Instead, they are processed through structured KDFs, which derive multiple independent keys for distinct protocol functions. In protocols such as MACsec, this includes keys for encryption, integrity protection, and key wrapping. This separation ensures that post-quantum or hybrid key establishment can be integrated into the control plane while preserving the symmetric key hierarchy required for efficient data-plane protection [64].

In virtualized environments, technologies such as Single Root I/O Virtualization (SR-IOV) allow multiple virtual machines to share a physical network interface through hardware-exposed Virtual Functions (VFs) while supporting traffic isolation [108, 109]. Post-quantum or hybrid security mechanisms can then be applied within these virtualized communication paths. In the evaluated virtualized deployments, post-quantum authentication and key establishment are executed in programmable software paths, while high-throughput symmetric data protection can be assigned to hardware offload components [9, 19].

Under these constraints, PQC changes protocol behavior primarily through larger cryptographic artifacts and operation-dependent computational costs. Optimized ML-KEM key establishment can be comparatively efficient, while signature operations, hybrid con-

structions, and constrained implementations may introduce greater processing cost. Larger public keys, ciphertexts, and signatures increase handshake size and can contribute to fragmentation, while protocol encapsulation such as ESP and SecTAG further reduces effective payload capacity [9, 40, 90].

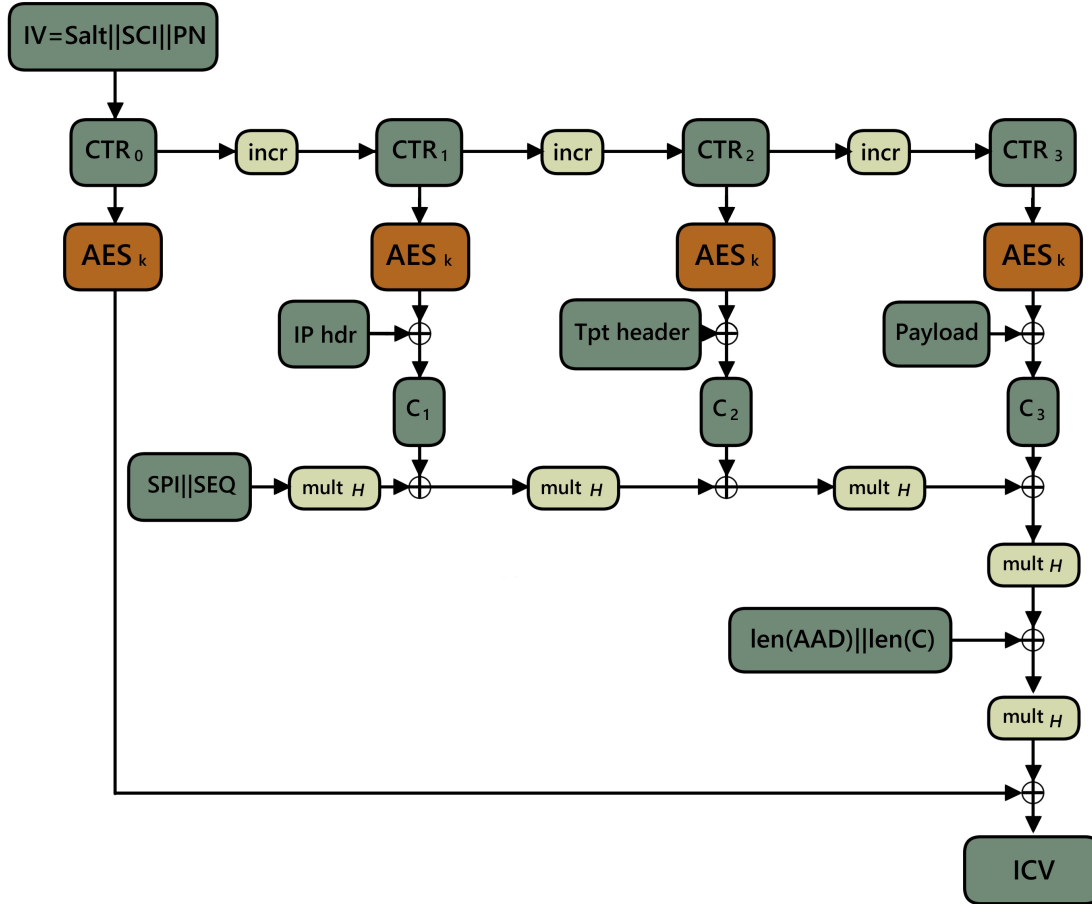


Figure 5.4: Use of AES-GCM for frame protection in MACsec. Adapted from Figure 6.2 of [1], with permission from the author.

Figure 5.4 illustrates how AES-GCM is applied within the MACsec protocol to provide authenticated encryption and how this process introduces protocol-level overhead. As shown in the figure, AES-GCM operates by generating a keystream using counter-mode encryption while simultaneously computing an authentication tag over both the encrypted payload and associated data, such as protocol headers [47].

This distinction between encrypted payload and authenticated associated data is essential in network protocols. Certain header fields must remain visible so that switches and routers can forward traffic correctly, but they must still be protected against tampering. AEAD constructions therefore allow protocols to preserve packet-processing functionality

while ensuring integrity and authenticity for both encrypted and non-encrypted protocol fields [52].

In the case of MACsec, this ensures that Ethernet frames are protected in terms of confidentiality and integrity, while the outer header remains visible for forwarding across the network. The figure highlights how additional inputs, including sequence numbers and security parameters, are incorporated into the encryption and authentication process. From a system perspective, this illustrates that even efficient symmetric mechanisms introduce structural overhead at the packet level, which reduces effective payload capacity and must be accounted for when post-quantum mechanisms further increase the cost of control-plane operations responsible for key establishment.

In the high-throughput deployments examined here, software-only data-plane processing can substantially reduce throughput [9, 14]. This established protocol distinction maps onto a layered execution model in practical systems. At the software level, post-quantum or hybrid authentication, policy negotiation, certificate validation, and key-establishment functions can be executed in programmable components. At the kernel level, systems manage cryptographic policies, routing decisions, and security associations. In high-performance environments, parts of packet processing may bypass the traditional kernel path through user-space frameworks, while selected operations may use hardware-assisted execution.

Continuous symmetric data protection can then be mapped to specialized hardware engines capable of line-rate encryption and authentication. This layered execution model preserves flexibility for evolving post-quantum mechanisms while accelerating high-volume traffic protection [14].

The deployment of PQC in real-world environments therefore introduces trade-offs that directly affect performance and scalability. Larger cryptographic artifacts increase bandwidth and memory pressure, while more complex operations increase processing cost. Different algorithms expose these costs differently. Some favor faster execution, while others reduce bandwidth consumption through smaller signatures [4]. In virtualized and edge environments, these trade-offs become more pronounced because isolation, resource sharing, and limited processing capacity further constrain deployment [10, 17, 18, 19]. Applied PQC must therefore balance security, performance, and adaptability across diverse deployment contexts.

Chapter 6

Applied Post-Quantum Cryptography in Practice

This chapter examines applied PQC, focusing on how quantum-resilient mechanisms behave when integrated into real communication and computing systems. Building on the system-level perspective developed in Chapter 5, the discussion moves from conceptual deployment challenges toward concrete architectural, protocol, and performance considerations. It first explains why PQC must be evaluated beyond algorithmic security alone, then analyzes how these mechanisms are incorporated across the networking stack. The chapter subsequently examines the performance and scalability constraints introduced by post-quantum schemes, followed by the role of hardware acceleration and cryptographic offloading in sustaining high-throughput communication. Finally, it considers the deployment of PQC in virtualized and multi-tenant environments, showing that practical adoption depends on the coordinated interaction between algorithms, protocols, hardware, and system architecture.

Source note. Selected architectural and protocol material in this chapter draws on Chapter 4 of [1] and Chapters 4–5 of [2]. The empirical analysis synthesizes the published experimental studies and independent measurements cited throughout.

6.1 From Cryptographic Algorithms to System Deployment

The transition from PQC as a theoretical construct to its deployment in real systems introduces challenges that extend beyond algorithmic design. In theoretical settings, cryptographic schemes are evaluated primarily according to mathematical hardness and resistance to known quantum algorithms. In practical environments, however, post-quantum schemes must also satisfy constraints imposed by protocols, software stacks, hardware platforms, and workload characteristics.

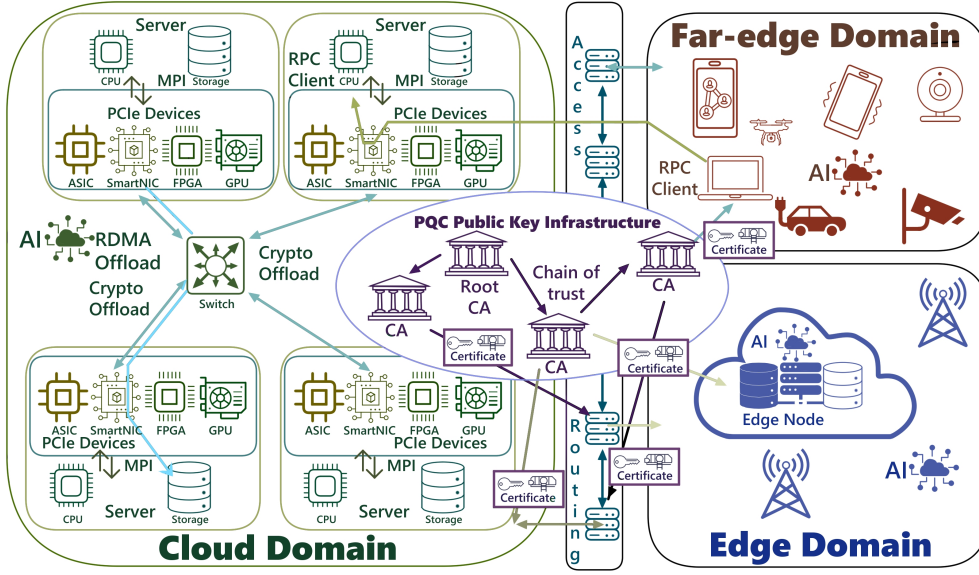


Figure 6.1: General architecture of a networked system with post-quantum cryptography integration across far-edge, edge, and cloud domains. Networking and security functions such as Remote Direct Memory Access (RDMA), Remote Procedure Call (RPC), and Message Passing Interface (MPI) may be supported by hardware components including Peripheral Component Interconnect Express (PCIe) devices, SmartNICs, and DPUs. Adapted from Figure 1 of [12], reprinted with permission from IEEE Proceedings.

One immediate consequence is a change in the computational and spatial profile of public-key cryptography. Depending on the scheme and implementation, post-quantum mechanisms may introduce additional computational cost, while their public keys, ciphertexts, and signatures are often larger than classical equivalents. These properties affect processing capacity, memory movement, packet size, bandwidth consumption, and latency. The suitability of a scheme therefore depends strongly on the deployment context rather than on theoretical security alone [36, 37, 40].

Applied PQC examines this interaction between algorithms and systems. It asks how quantum-resilient mechanisms behave when integrated into real communication protocols, high-speed packet-processing pipelines, hardware-assisted execution paths, and virtualized environments. This perspective is essential because a scheme that is secure in isolation may still become difficult to deploy if its computational or communication costs exceed the capabilities of the target infrastructure.

This architectural transition is illustrated in Figure 6.1, which presents a generalized view of a modern networked system in which PQC is integrated across far-edge, edge, and cloud domains. The figure highlights how post-quantum certificates and key-establishment mechanisms support trust across distributed infrastructure, while networking and security functions may be assisted by hardware components close to the data path. At the same time, it reflects the increasing distribution of computation across the infrastructure, where key networking and security functions are no longer confined to the host processor.

6.2 Post-Quantum Cryptography in the Networking Stack

Cryptography in modern systems is not an isolated component, but a function embedded across the communication stack. Its placement is commonly described through layered models such as the OSI model and the TCP/IP architecture, where security mechanisms are integrated at different levels depending on the protocol [110, 111]. At the data link layer, mechanisms such as MACsec protect Ethernet frames directly by adding integrity and confidentiality fields while preserving addressing information for forwarding [52]. Because MACsec operates at the link layer, it is transparent to higher-layer protocols and does not require changes to the IP, transport, or application layers. Once supported by the communicating endpoints, protected Ethernet frames can traverse intermediate network elements while preserving the existing communication stack. At the network layer, IPsec secures IP packets by encapsulating and protecting their contents, providing end-to-end security between hosts and network domains [62]. Above the transport layer, TLS protects application data by encapsulating it in encrypted records [50]. Because these protocols operate at different layers, their suitability for hardware acceleration also differs. Application-layer mechanisms such as TLS are often implemented primarily in software, whereas lower-layer mechanisms such as IPsec and MACsec can more directly exploit network-interface and data-path acceleration. Within this layered view, cryptography is executed continuously as part of packet processing, becoming an integral part of the communication pipeline rather than an external operation.

In high-performance environments, this integration can extend further through kernel-bypass frameworks such as DPDK. Instead of relying on interrupt-driven packet processing within the operating system, DPDK enables user-space applications to access network interface hardware directly using polling-based mechanisms. This avoids overhead associated with context switching, socket buffers, and kernel networking layers, allowing packets to be processed at significantly higher throughput. In such architectures, cryptographic operations, including post-quantum handshakes and packet protection, can be integrated directly into the user-space data path, making them part of a highly optimized packet-processing pipeline.

This layered placement also interacts with traffic patterns. TLS is widely used for application-layer client–service and service–service communication, while IPsec and MACsec protect traffic at the network and link layers across both east–west and north–south paths. The experimental deployments considered here illustrate both traffic patterns. Intra-data-center east–west communication can be protected through DPU-offloaded AES-256-GCM at 100 Gbit/s line rate, while north–south communication between a wireless edge client and a cloud-side DPU can be protected through a post-quantum-secured IPsec tunnel at approximately 0.486 Gbit/s [10, 17].

Such dual traffic patterns are particularly relevant for confidential AI workloads, where external clients may access cloud-hosted models through north–south traffic while distributed training or inference pipelines generate large volumes of protected east–west communication inside the data center [17].

The complementary roles of MACsec and IPsec also make it possible to construct multilayer quantum-resilient data-transfer architectures. In such design, MACsec provides link-layer protection for Ethernet frames within local or data-center networks, while IPsec provides network-layer protection for IP packets across routed or potentially untrusted domains. Deploying both mechanisms together creates a defense-in-depth model in which data is protected at multiple levels rather than relying on a single cryptographic layer. This is particularly relevant for high-capacity data-center and wide-area deployments, where traffic may traverse both trusted local fabrics and external network paths [16].

This distinction also extends beyond centralized data centers to cloud-edge continuum environments, where cloud servers, edge nodes, and constrained devices exchange data across distributed infrastructure. In such settings, DPUs can act as intermediate security and processing nodes, performing encryption, packet handling, and post-quantum protocol operations close to the data path. This reduces the burden on central High-Performance Computing (HPC) or cloud servers while enabling quantum-resilient communication between edge and cloud resources [11].

Building on the control-plane/data-plane separation introduced in Chapter 2, the post-quantum transition primarily affects the authentication and key-establishment mechanisms of existing secure protocols rather than continuous symmetric traffic protection. In MACsec, this concerns the MKA protocol; in IPsec, IKEv2; and in TLS, the handshake protocol [50, 60, 64]. The following discussion examines how these protocol-specific mechanisms are adapted for post-quantum and hybrid key establishment and how the resulting changes affect key derivation, message size, and deployment behavior.

This transformation can be better understood by examining how session keys are derived in classical protocol designs. In IKEv2, shared secrets are established through DH KEX and subsequently expanded using pseudo-random functions to produce multiple cryptographic keys for encryption, integrity protection, and authentication.

In this process, a shared secret is first derived from the exchanged key material and then expanded into a set of session keys using a key derivation function. These derived keys are used for both control-plane protection (e.g., authentication and integrity of handshake messages) and for generating data-plane keys used by protocols such as ESP.

From a post-quantum perspective, this construction introduces a fundamental vulnerability because the initial shared secret depends entirely on classical DH assumptions, which can be efficiently broken by quantum algorithms such as Shor’s. As a result, all derived session

keys inherit this weakness, meaning that both control-plane and data-plane security can be compromised if the underlying key exchange is not replaced.

More specifically, the insecurity of classical IPsec under quantum adversaries arises from two independent factors. First, a recorded classical DH exchange can be retrospectively compromised by a sufficiently capable quantum adversary, so classical forward secrecy does not protect that session against a future quantum attack [22]. Second, authentication mechanisms based on classical digital signatures, such as elliptic-curve cryptography, can also be compromised, allowing an adversary to impersonate legitimate parties. While authentication is less directly affected by store-now-decrypt-later attacks, its compromise still undermines trust establishment in the protocol. Together, these limitations demonstrate that both key exchange and authentication must be redesigned to achieve quantum resilience.

The practical implications become clearer when examining specific protocols. In the case of IPsec, the transition to PQC redefines the Internet Key Exchange process, where shared secrets are negotiated between communicating parties. Classical key exchange mechanisms are augmented or replaced by post-quantum alternatives. In many cases, these are deployed within hybrid constructions that combine multiple sources of key material. This transition is supported by extensions such as RFC 9370, which enables multiple successive key exchanges within IKEv2 to accommodate post-quantum primitives [13, 102].

In these hybrid constructions, session keys are derived through cascade-based approaches, where key material from multiple exchanges is combined iteratively. A concrete realization of this approach is defined in RFC 9370, which extends IKEv2 to support multiple sequential key exchanges within a single session [102]. In this model, each additional exchange contributes new entropy to the evolving cryptographic state.

This model also enables more advanced IPsec configurations in which classical key exchange, post-quantum key encapsulation, and QKD-derived key material all contribute to the same IKEv2 negotiation. In such triple-hybrid settings, mechanisms such as X25519, ML-KEM-1024, and QKD can be combined during authenticated key exchange, while the resulting key material is subsequently used to establish ESP security associations for data-plane protection [8, 13].

Rather than relying on a single shared secret, the protocol updates the session key material after each exchange, incorporating classical DH, post-quantum key encapsulation mechanisms, and potentially QKD. When the successive exchanges and key schedule are composed securely, the design diversifies the security dependencies across classical, post-quantum, and QKD-derived material [13, 102]. Once established, these keys are used by the ESP mechanism, which constitutes the data-plane component of IPsec. While IKEv2 operates in the control plane to establish trust and derive key material, ESP is responsible for the continuous protection of data traffic through symmetric encryption, typically relying on AES-256-GCM [13, 51]. The data plane itself remains largely unchanged, while

control-plane negotiation becomes more complex due to larger key material and additional processing steps required to derive shared secrets.

Alternative approaches based on pre-shared post-quantum keys have also been proposed. In these designs, a post-quantum pre-shared key is incorporated into the key derivation process prior to or alongside classical exchanges. While this method simplifies deployment and avoids major protocol modifications, the pre-shared post-quantum contribution does not by itself provide post-quantum forward secrecy; protection depends on the long-term confidentiality and lifecycle of the pre-shared material [103]. These trade-offs highlight the importance of dynamic hybrid key-exchange mechanisms, such as those defined in RFC 9370, for practical post-quantum deployments [13, 102].

The same issue appears in practical IPsec prototypes when native post-quantum rekeying is not yet supported by the control-plane implementation. If rekeying falls back to classical mechanisms, the renewed session may no longer provide quantum-resilient security. One practical workaround is to disable classical rekeying and rerun the post-quantum authentication and key-establishment procedure whenever fresh key material is required. This highlights that post-quantum migration must address not only initial session establishment, but also key renewal and long-lived security-association management [10].

Independent implementations indicate that ML-KEM-based IPsec is also progressing beyond experimental cryptographic prototypes. Cloudflare made hybrid ML-KEM key establishment for its IPsec service generally available in 2026 using the emerging Internet Engineering Task Force (IETF) IKEv2 profile and reported interoperability with multiple commercial implementations [112]. Current Apple operating systems also expose ML-KEM-768 and ML-KEM-1024 as additional post-quantum key-exchange methods for native IKEv2 VPN configurations [113]. Similarly, current Cisco routing platforms support hybrid ML-KEM key establishment in IKEv2 and explicitly account for the fragmentation risks associated with larger ML-KEM parameters [114].

Independent experimental results provide a complementary performance reference. In an open O-RAN testbed using strongSwan and liboqs, ML-KEM-based IKEv2/IPsec increased tunnel-establishment latency by approximately 2.7–4.7 ms compared with classical ECDH-based IPsec, while the protected control application remained stable [115]. The current IETF ML-KEM/IKEv2 specification likewise treats message size and fragmentation as deployment considerations because higher-parameter ML-KEM public values may exceed common network packet payload sizes [116]. These results support the practical feasibility of post-quantum IPsec while showing that interoperability, packet size, and handshake integration remain relevant system-level constraints.

A practical implementation path for post-quantum IPsec is the integration of post-quantum cryptographic plugins into existing IPsec software stacks. One example is the integration of the Open Quantum Safe (OQS) library into the accelerated Mellanox strongSwan stack for DPUs. In this approach, post-quantum authentication is provided through schemes such

as ML-DSA or FN-DSA, post-quantum key establishment is provided through ML-KEM, and the resulting session keys are used to protect data-plane traffic with AES-256. This demonstrates that post-quantum IPsec can be introduced through extensions to existing protocol implementations, rather than by replacing the entire networking stack [11].

A complementary deployment approach is to perform post-quantum authentication and key establishment outside the native IPsec negotiation and then configure the IPsec tunnel using the pre-shared key method. In this model, an authenticated OpenSSL channel is first established, post-quantum signatures such as FN-DSA or ML-DSA are exchanged, ML-KEM is used to derive post-quantum key material, and the classical and post-quantum keys are mixed before being installed as the key material for an AES-256-GCM-protected IPsec tunnel. This shows that quantum-resilient IPsec can be experimentally realized even before full native support for post-quantum IKEv2 is available [10].

This type of integration illustrates that the transition to PQC affects more than the cryptographic primitive itself. Supporting post-quantum algorithms inside IPsec requires changes across several parts of the software stack, including algorithm identifiers, certificate and key handling, key-exchange logic, plugin interfaces, and tools for generating post-quantum certificates. For example, ML-KEM-based key establishment must be connected to the key-exchange component of the protocol implementation, while ML-DSA, FN-DSA, and SLH-DSA require support in the certificate and signature-handling path. As a result, post-quantum migration becomes a protocol-engineering and software-integration task, not merely a matter of linking a new cryptographic library [11].

To make this behavior more concrete, Figure 6.2 illustrates the structure of an IPsec-protected packet and the role of the ESP in both transport and tunnel modes.

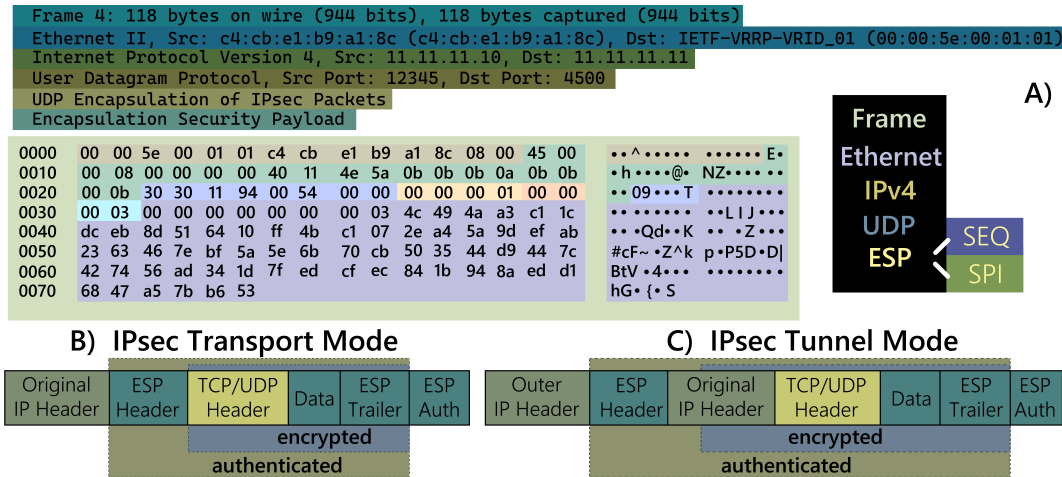


Figure 6.2: IPsec packet structure and ESP encapsulation in transport and tunnel modes. Adapted from Figure 4.7 of [2], with permission from the author.

Figure 6.2 shows a Wireshark capture of an IPsec-protected packet transmitted over Ethernet using User Datagram Protocol (UDP) encapsulation, together with schematic representations of ESP in transport and tunnel modes. The ESP header introduces fields such as Security Parameters Index (SPI), which identifies the corresponding security association, and the Sequence Number (SEQ), which provides replay protection.

In transport mode, the original IP header remains visible, while the payload is encrypted and authenticated. In tunnel mode, the entire original IP packet is encapsulated within the ESP payload and protected, with an additional outer IP header used for routing. This distinction highlights that IPsec modifies packet structure at the network layer while preserving forwarding functionality.

From a post-quantum perspective, the figure also illustrates an important separation. While ESP continues to rely on efficient symmetric encryption for data-plane protection, the mechanisms used to establish the underlying security associations, such as IKEv2, are the components that must be adapted to support post-quantum or hybrid key exchange schemes.

In Linux-based deployments, this separation is reflected in the way IPsec is implemented through the Transform Framework (XFRM) framework. Packets are matched against installed policies and then processed according to the corresponding Security Associations. Hardware support can be introduced at different levels. In crypto offload mode, the network interface performs encryption and decryption, while the kernel remains responsible for the remaining packet-processing steps. In packet offload mode, a larger part of the IPsec data path is delegated to the network interface, including encryption, decryption, ESP encapsulation and decapsulation, and the handling of synchronized policies and Security Associations. This distinction shows that offloading is not a single mechanism, but a spectrum of architectural choices that determines how much responsibility remains with the host operating system.

This distinction also affects how different transport protocols behave under post-quantum-secured IPsec. Experimental results with PQC-enabled IPsec over high-speed DPU links show that Transmission Control Protocol (TCP) and UDP do not benefit equally from offloading. In host-to-host NIC-mode experiments, packet offload improved UDP throughput more significantly than crypto-only offload, because the hardware handled both encryption and ESP packet processing. In DPU-mode virtualized network-slicing experiments, UDP scaled particularly well across multiple VFs, while TCP remained constrained by connection management, acknowledgements, congestion control, and interaction with encapsulation overhead. This shows that post-quantum-secured networking depends not only on the cryptographic primitive and offload capability, but also on transport-layer behavior and packet-processing semantics [18].

The same pattern appears in MACsec, where the focus lies on secure communication at the link layer. Here, PQC affects the key management framework responsible for distributing

encryption keys among participating devices. The MKA protocol is extended to support quantum-resilient key establishment, enabling the integration of post-quantum primitives within existing Ethernet security frameworks. In practical deployments, initial keying material may be derived through higher-layer protocols and subsequently expanded through a hierarchical key derivation process. In this hierarchy, a MSK is used to derive a CAK and CKN, which are then processed by MKA to generate the SAK used for frame protection. Additional keys, including the KEK and ICK, support secure key distribution and integrity protection for MKA control messages.

In the quantum-resilient MACsec configuration considered here, 256-bit symmetric mechanisms are used throughout the relevant protection and key-management functions, including AES-256-GCM for data-plane protection and corresponding 256-bit mechanisms for key wrapping and key derivation. The initial shared secret can then be established through a post-quantum or hybrid mechanism and incorporated into the MKA key hierarchy [14, 64].

While the encryption of Ethernet frames continues to rely on efficient symmetric mechanisms, the introduction of post-quantum primitives modifies how these keys are generated and managed. At the same time, performance requirements remain strict. To sustain line-rate operation, cryptographic processing in the data plane is typically offloaded to specialized hardware engines, such as those available in modern DPUs. In a quantum-secure MACsec deployment, an extended TLS-based control-plane exchange can first be used to authenticate peers and establish a hybrid MSK from classical, post-quantum, and QKD-derived key material. This MSK is then expanded through the MKA hierarchy into the CAK, CKN, KEK, ICK, and finally the SAK, which is used for AES-GCM-based frame protection. In this architecture, the control plane provides crypto-agile quantum-secure key establishment, while the MACsec data plane is offloaded to the DPU, enabling line-rate encrypted Ethernet communication in the 200 Gbit/s regime [14].

At the transport layer, TLS provides another important example of how PQC is integrated into communication protocols. TLS operates through two main components, the handshake protocol, which is responsible for authentication and key establishment, and the record protocol, which protects application data [50]. The handshake becomes the primary integration point for post-quantum mechanisms. Classical key-establishment and authentication mechanisms are replaced or combined with post-quantum alternatives, including ML-KEM for key establishment and ML-DSA for digital signatures [13].

In triple-hybrid TLS 1.3 deployments, the handshake can be extended so that classical key exchange, post-quantum key encapsulation, and QKD-derived key material all contribute to the final session keys. A practical architecture triggers the classical, post-quantum, and QKD key-negotiation procedures on the client side before the ClientHello message is sent. This design allows the client to include the classical public key, the post-quantum key-share information, and the QKD key identifier in the initial handshake message, while avoiding server-side QKD key exhaustion caused by unsolicited key retrieval requests [13].

In hybrid TLS constructions, shared secrets derived from multiple mechanisms are combined to produce the final session keys. This approach ensures that communication remains secure as long as at least one of the underlying cryptographic assumptions holds. In triple-hybrid TLS 1.3, this can be implemented through a concatenation-based approach in which the shared secrets produced by classical key exchange, post-quantum key encapsulation, and QKD are concatenated and provided as input key material to the TLS key schedule. For example, three 32-byte secrets can be combined into a 96-byte input to the HKDF-based key derivation process, from which handshake traffic secrets, application traffic secrets, write keys, and initialization vectors are derived [13]. Although the structure of the protocol remains similar, the increased size and complexity of post-quantum artifacts affect handshake behavior and communication efficiency. Once the handshake is completed, the TLS record layer continues to rely on symmetric encryption, typically AES-GCM, which can be accelerated in hardware to maintain high throughput. In practice, this separation allows control-plane operations to incorporate post-quantum security while preserving efficient data-plane performance [13].

These developments show that PQC reshapes how security is implemented across the networking stack. Rather than introducing entirely new protocols, the transition builds on existing ones by modifying mechanisms at multiple layers, especially in the control plane. These modifications affect how keys are established, how identities are verified, and how secure sessions are created, while the data plane continues to rely on efficient symmetric cryptography. As a result, the networking stack becomes a key constraint in deploying PQC, requiring coordinated changes across protocols and system components.

6.3 Performance and Scalability in Post-Quantum Systems

The move from architectural integration to real-world deployment makes performance a central concern in PQC. Whereas previous sections examined how these mechanisms are integrated into systems and protocols, their impact becomes more evident when evaluated in terms of computational cost, communication overhead, and system scalability. In this context, PQC is not only a security mechanism, but also a determining factor in overall system behavior, directly influencing latency, throughput, and resource usage.

Importantly, these effects must be understood within the constraints of existing packet-processing pipelines. In modern systems, packets traverse highly optimized execution paths across network interfaces, kernel or user-space stacks, and transport protocols. Post-quantum mechanisms must integrate into this pipeline without disrupting its structure. As a result, any additional computational or communication overhead introduced by PQC must be absorbed within an already tightly constrained processing model, reinforcing that deployment feasibility is determined at the system level rather than by algorithmic efficiency alone.

Post-quantum mechanisms have computational profiles that differ substantially from classical public-key cryptography, but the resulting overhead depends on the algorithm, operation, parameter set, implementation, and hardware architecture. Optimized ML-KEM operations can be comparable to, and on some platforms faster than, classical elliptic-curve key establishment, whereas post-quantum signatures, hybrid constructions, and resource-constrained implementations may introduce more substantial costs [40, 90, 92]. Consequently, computational cost should be evaluated per operation and deployment environment rather than treated as a uniform property of PQC.

This variation is also visible in the measurements considered in this chapter. For example, certain digital signature operations in ML-DSA can require on the order of 10^7 computational cycles, while key encapsulation mechanisms such as ML-KEM include subroutines that reach approximately 5×10^5 cycles [3].

A dominant computational bottleneck in lattice-based constructions arises from hashing, particularly Keccak-based functions such as SHAKE. In practical software implementations, hashing can account for up to approximately 85% of the total execution time, making it the primary contributor to computational cost in schemes such as ML-DSA [15].

The published evaluation in [15] examines alternative byte-generation approaches for the CRYSTALS-Dilithium implementation on Advanced RISC Machine (ARM)v8, using SHAKE as the performance baseline and comparing it with alternatives including TurboSHAKE, SHA-2, AES-CTR, and ASCON. The results show that the relative performance of these constructions changes substantially when architecture-specific acceleration is available. In particular, SHA-2 and AES-CTR perform poorly relative to SHAKE in their reference implementations, but benefit considerably from the cryptographic instruction-set extensions available on the evaluated ARMv8 platform.

Because the published experiment uses the pre-standardization CRYSTALS-Dilithium implementation, the measurements are reported using the original Dilithium2, Dilithium3, and Dilithium5 parameter-set names rather than being relabeled as measurements of the finalized ML-DSA standard. Table 6.1 summarizes the hardware-optimized SHA-2 and AES-CTR results that are most relevant to the architecture-dependent performance comparison.

The optimized measurements show that the benefit of alternative byte-generation mechanisms depends strongly on the available hardware support. Across the three evaluated parameter sets, optimized AES-CTR reduces the cycle count relative to the SHAKE baseline for key generation, signing, and verification, with the largest measured reductions reaching approximately 44% for key generation and 39% for verification. Hardware-optimized SHA-2 also reduces the cycle count for key generation and verification, but remains slightly more expensive than SHAKE for signature generation [15].

Table 6.1: ARM CPU-cycle comparison for optimized Dilithium operations using SHAKE as the baseline and hardware-optimized SHA-2 and AES-CTR implementations. Percentage values indicate the change in cycle count relative to the SHAKE baseline; negative values indicate fewer cycles. Data reported in [15].

Parameter set	Operation	SHAKE baseline	SHA-2 optimized	Δ SHA-2 vs. SHAKE	AES-CTR optimized	Δ AES-CTR vs. SHAKE
Dilithium2	Key generation	36,870	34,194	−7.29%	23,196	−37.13%
	Signature	130,224	135,564	+4.07%	107,943	−17.12%
	Verification	40,920	35,601	−12.99%	28,833	−29.53%
Dilithium3	Key generation	65,880	59,190	−10.13%	40,656	−38.24%
	Signature	220,683	231,513	+4.91%	183,495	−16.85%
	Verification	65,259	53,373	−18.23%	43,026	−34.06%
Dilithium5	Key generation	100,827	87,102	−13.14%	56,220	−44.23%
	Signature	275,829	280,860	+2.06%	218,622	−20.77%
	Verification	106,173	88,083	−17.15%	64,719	−39.04%

Beyond hashing, additional overhead arises from operations such as the NTT used for polynomial multiplication, as well as from sampling procedures based on probability distributions [3].

These measurements show that architecture-specific primitive choices can materially affect research implementations. They should not, however, be interpreted as permission to substitute internal primitives in finalized ML-DSA, whose approved construction is fixed by FIPS 204 [15, 37]. More broadly, the results demonstrate the importance of evaluating implementation choices against both performance targets and the requirements of the applicable standard.

This implementation-dependent behavior is also visible in key-establishment performance. Although post-quantum mechanisms generally introduce larger artifacts and more complex operations, optimized lattice-based KEMs can be competitive with classical key exchange on suitable hardware. In the pre-standardization DPU-based evaluations reported in [9], optimized CRYSTALS-Kyber implementations running on ARM cores achieved substantially higher key-establishment throughput than the corresponding reference implementations and, in some cases, outperformed optimized elliptic-curve key exchange. For example, Kyber-512 reached up to approximately 25,000 encapsulation or decapsulation operations per second, compared with roughly 5,000 key exchanges per second for ECDH P-256, while Kyber-1024 reached approximately 10,000 encapsulations per second. These measurements characterize the implementation lineage that subsequently led to ML-KEM, rather than a benchmark of the finalized FIPS 203 implementation.

In particular, achieving line-rate secure communication with post-quantum mechanisms requires careful alignment between cryptographic primitives and the underlying execution architecture, including processor instruction sets and memory hierarchies [15].

These costs are further shaped by hardware characteristics. On resource-constrained platforms, limited processor and memory resources can increase latency, although the magnitude depends strongly on the implementation and on the availability of instruction-set acceleration [15, 17].

Parallelization strategies, such as GPU-based batching, can yield substantial performance improvements, but their effectiveness depends strongly on workload structure. Experimental evaluation of ML-KEM and ML-DSA on a heterogeneous DPU shows that single-operation GPU execution can be slower than execution on embedded ARM cores, since kernel-launch overhead dominates when only one message is processed. However, when operations are aggregated into batches, this overhead is amortized. Batch sizes of ten operations already reverse the performance trend, while batches of 100 and 1000 expose the full benefit of the GPU-accelerated architecture [21].

At the largest evaluated batch size, the heterogeneous CPU–GPU configuration achieved total end-to-end speedups of approximately $30\times$ – $45\times$ for ML-KEM and $55\times$ – $94\times$ for ML-DSA. Operation-level improvements were likewise substantial, with key generation, encapsulation, decapsulation, signing, and verification benefiting from large-scale parallel execution. These results show that GPU-enabled DPUs are particularly effective for high-throughput post-quantum services where many identities, signatures, or key-establishment operations must be processed concurrently [21].

Independent system-level measurements reinforce the need to distinguish cryptographic microbenchmarks from end-to-end deployment behavior. Rios and Montenegro evaluate post-quantum mechanisms across standalone cryptographic operations, TLS handshakes, and complete VPN connection establishment on both ARM and x86 systems. Their results show that architecture-specific performance advantages observed for isolated primitives can be substantially reduced once protocol processing is included, and can become secondary when network delay or packet loss is introduced [40]. In particular, ML-KEM exhibits comparatively modest primitive-level overhead, while signature operations and protocol processing can contribute a larger share of total connection-establishment cost. This independently supports evaluating post-quantum cryptography across multiple system layers rather than extrapolating deployment performance directly from primitive execution time.

Alongside computational overhead, PQC results in a marked increase in communication cost due to the larger size of cryptographic artifacts. Public keys, signatures, and ciphertexts are considerably larger than their classical counterparts. For instance, ML-DSA public keys can be approximately twenty times larger than those of ECDSA, while FN-DSA signatures, although more compact than ML-DSA signatures, still introduce larger cryptographic artifacts than many classical signature schemes [6, 9].

Post-quantum authentication presents a different deployment profile from key establishment. Independent TLS measurements report that the computational overhead of standardized

post-quantum mechanisms can be comparable with classical alternatives, while the larger amount of cryptographic data transmitted during the handshake remains a significant constraint [90]. Operational analysis at Internet scale reaches a similar conclusion. Cloudflare estimates that a straightforward substitution of ML-DSA-44 throughout a conventional WebPKI authentication chain would add on the order of 15 kB to a TLS connection, considerably more than the additional data introduced by hybrid ML-KEM key establishment [117]. This distinction helps explain why post-quantum key agreement has reached production deployment more rapidly than post-quantum certificate authentication and reinforces the need to evaluate key establishment and signatures separately.

The overhead introduced by QKD has a different profile. In triple-hybrid TLS and IPsec deployments, QKD contributes very little to packet-size overhead because the protocol mainly needs to transmit a key identifier rather than the key itself. In the evaluated TLS implementation, the QKD key identifier required only 36 bytes, making its bandwidth impact negligible compared with the larger public keys and ciphertexts introduced by post-quantum mechanisms. The main cost of QKD instead appears as latency during key retrieval from the key management system. In the evaluated TLS deployment, real QKD equipment introduced approximately 48 ms of additional key-retrieval overhead, with the overall added handshake cost reported at approximately 57 ms [13].

The feasibility of QKD-assisted protocols also depends on the availability of fresh key material. In the evaluated system, the QKD equipment provided an average secret key rate of approximately 2.5 kbit/s, so the rate at which fresh key material can be supplied becomes an additional deployment constraint. For IPsec, the key-rate requirement is shaped by the key-usage limits of AES-256-GCM. Assuming a 379 GB usage limit per key and a 100 Gbit/s encrypted tunnel, fresh key material should be introduced approximately every 30 s. Under the same 2.5 kbit/s secret key rate, the evaluated system provides approximately nine 256-bit keys per second, which is sufficient for the reported high-throughput IPsec scenario [13].

This overhead can be mitigated in repeated TLS connections through the existing resumption mechanisms of TLS 1.3. Since the key schedule derives a resumption master secret, subsequent sessions can benefit from the security of the previously established triple-hybrid handshake without repeating the full key-negotiation procedure. This makes triple-hybrid TLS more practical in scenarios where clients establish multiple connections to the same server, because the initial latency cost does not necessarily apply to every subsequent session [13].

Production measurements show that the observable cost of post-quantum TLS depends strongly on connection behavior. In an AWS KMS benchmark, hybrid X25519MLKEM768 reduced request throughput by approximately 2.3% when every request forced a new TLS connection, from 108.7 to 106.2 transactions per second. With TLS connection reuse enabled, however, the measured difference fell to approximately 0.05%, from 216.1 to 216.0

transactions per second [96]. This demonstrates how session reuse can amortize the cost of post-quantum key establishment in long-lived or repeated application workloads.

The magnitude of this overhead changes considerably in constrained environments. An independent 2026 evaluation of MQTT over TLS on an emulated ARM64 platform found that a hybrid ML-KEM-768 and ML-DSA-65 configuration increased connection-establishment time by approximately $4.7\times$ relative to its classical TLS baseline under the evaluated conditions. Once the secure connection had been established, however, application-message publication latency remained negligible, leading the authors to identify persistent sessions as an effective way of amortizing the post-quantum handshake cost [118]. This result further illustrates why post-quantum overhead should be associated with specific workloads and connection lifetimes rather than generalized across an entire communication session.

Communication effects can nevertheless remain visible even when the underlying key-encapsulation computation is efficient. During the earlier large-scale Chrome deployment of hybrid Kyber key establishment, the approximately kilobyte-sized post-quantum key share caused the TLS ClientHello to span an additional packet and was associated with an approximately 4% increase in median desktop handshake latency [119]. More recent Apple deployments with standardized X25519MLKEM768 independently expose the same interoperability issue at the protocol level: Apple documents connection failures with some legacy TLS servers that incorrectly process the larger ClientHello [95]. These observations show that packetization and compatibility effects can become deployment bottlenecks even when the cryptographic operation itself is computationally inexpensive.

This increase in size directly affects bandwidth consumption and may lead to packet fragmentation during protocol handshakes, since cryptographic payloads can exceed standard packet size limits. At the protocol level, additional encapsulation overhead consequently reduces the effective payload capacity. In MACsec, Ethernet frames are extended with security fields such as the SecTAG and ICV, reducing the available space for application data [52]. Similarly, IPsec introduces ESP headers and trailers that modify packet structure and decrease usable bandwidth [9, 61].

In high-speed environments, larger frames such as jumbo frames can mitigate per-packet overhead and improve effective throughput in suitable deployments [10, 18].

The impact of transport-layer behavior is particularly visible in high-speed IPsec deployments. In PQC-secured IPsec experiments over 400 Gbit/s DPU-based links, TCP throughput remained significantly below the unencrypted baseline, even when offloading was enabled. By contrast, UDP achieved substantially higher throughput, especially when packet offload was available. This difference shows that the performance of post-quantum-secured communication is shaped not only by cryptographic computation, but also by transport-layer mechanisms such as connection management, congestion control, acknowledgements, and packet-size adaptation [18].

At the data-plane level, the impact of packet size is also visible in in-line encrypted IPsec links. Experimental results with DPU-based AES-256 encrypted flows show that encrypted traffic can approach line-rate performance across different Maximum Transmission Unit (MTU) sizes and thread configurations. Smaller MTU values can be handled efficiently by the DPU, while larger MTU configurations introduce a modest throughput penalty, approximately 10% compared with clear transmission and about 15% relative to the channel capacity. This indicates that packet-level configuration remains an important factor in preserving throughput under post-quantum-secured communication [9].

Jointly, these computational and communication costs exert a direct impact on system performance, particularly in terms of throughput and latency. Without hardware acceleration, post-quantum operations can severely degrade network performance, reducing throughput to only a small fraction of the available link capacity. In the evaluated very-high-throughput IPsec and MACsec systems, combined processing and communication costs substantially reduced throughput when data-plane protection remained in software. Host-based post-quantum-secured IPsec fell below 5 Gbit/s, while a DPU without cryptographic acceleration reached approximately 13.3 Gbit/s on a 100 Gbit/s link; software-based MACsec likewise used only a small fraction of the tested 200 Gbit/s channel [7, 14].

Latency is likewise affected, as the increased computational complexity of post-quantum handshakes introduces additional delay during session establishment. By contrast, when these operations are offloaded to specialized hardware components such as DPUs, performance improves markedly. In an end-to-end optical fiber experiment, hardware-accelerated DPU offloading sustained 92.3 Gbit/s for a post-quantum-secured IPsec tunnel, compared with a 98.2 Gbit/s unencrypted baseline. In contrast, the same tunnel reached only 13.3 Gbit/s on a DPU without hardware acceleration and less than 5 Gbit/s when processed on the host server. These observations demonstrate that hardware support can be critical for preserving high throughput in very-high-throughput post-quantum deployments, while its necessity remains dependent on the target workload and system architecture [7].

A further IPsec evaluation shows that this performance behavior depends strongly on the separation between the control plane and the data plane. When IKEv2 was extended with individual, hybrid, and triple-hybrid authenticated key exchange mechanisms based on X25519, ML-KEM-1024, and QKD, the negotiation phase experienced different latency profiles. In particular, incorporating QKD key material introduced an additional delay of approximately 35 ms, mainly due to key retrieval through the Key Management System (KMS). However, this control-plane overhead had negligible impact on data-plane throughput, which remained close to line rate because bulk traffic was protected using hardware-accelerated AES-256-GCM on the DPU. The experiment achieved 91.1 Gbit/s encrypted throughput, showing that even complex triple-hybrid key establishment can be practical when its cost is isolated from the high-throughput packet-processing path [8].

The same result also shows that key renewal can be treated as a runtime policy variable rather than as a fixed protocol parameter. By adapting the Key Renewal Interval (KRI) according to the availability of QKD key material, the status of the quantum channel, or the volume of encrypted traffic, systems can increase the frequency of rekeying without necessarily reducing data-plane throughput. In the evaluated DPU-based IPsec deployment, renewal intervals as low as 1 second were supported without disrupting encrypted data transmission, provided that sufficient key material was available [8].

This behavior has been demonstrated in an end-to-end optical fiber transmission scenario, where a post-quantum-secured IPsec tunnel was established between two independent servers using DPUs for cryptographic processing. The experimental setup replicated an inter-data-center communication environment, with two DPU-equipped servers connected through an optical network and a 240 km fiber link. In this configuration, post-quantum authentication and key exchange were integrated into the tunnel establishment process, while high-throughput data transmission was protected using symmetric encryption. The experiment demonstrated that post-quantum security can be incorporated into realistic high-speed optical communication systems when cryptographic processing is offloaded to appropriate hardware [7].

Figure 6.3 provides experimental evidence of these effects by comparing throughput and CPU utilization under different deployment configurations for post-quantum IPsec tunnels. The evaluated scenarios include unencrypted communication, server-based PQC processing, and DPU-based execution with and without hardware acceleration.

The throughput results (left) show that unencrypted communication achieves near line-rate performance, serving as a baseline. When post-quantum IPsec processing is offloaded to a DPU with hardware acceleration, throughput remains close to this baseline, demonstrating that hardware-assisted execution can effectively absorb the additional cost introduced by post-quantum mechanisms. In contrast, when hardware acceleration is not available, throughput drops significantly, even when processing is offloaded to the DPU. The degradation becomes more pronounced when cryptographic operations are executed entirely on the server, where throughput falls to only a small fraction of the available link capacity.

The CPU utilization results (right) further highlight the system-level implications. When post-quantum IPsec processing is performed on the server, CPU usage remains high across all packet sizes, exceeding 80% in the evaluated configuration and indicating substantial computational pressure and limited scalability. By contrast, offloading cryptographic operations to the DPU reduces CPU utilization to approximately 20%, freeing around 60% of host processing capacity for application workloads [7].

For the evaluated very-high-throughput IPsec configuration, hardware offloading was necessary to approach line-rate encrypted throughput while substantially reducing host CPU utilization. This conclusion is specific to the tested architecture and workload. Independent TLS, VPN, embedded, and lower-rate IPsec evaluations show that post-

quantum key establishment itself can introduce comparatively modest overhead in other environments, where protocol processing, communication effects, or network conditions may instead dominate system performance [40, 92, 115].

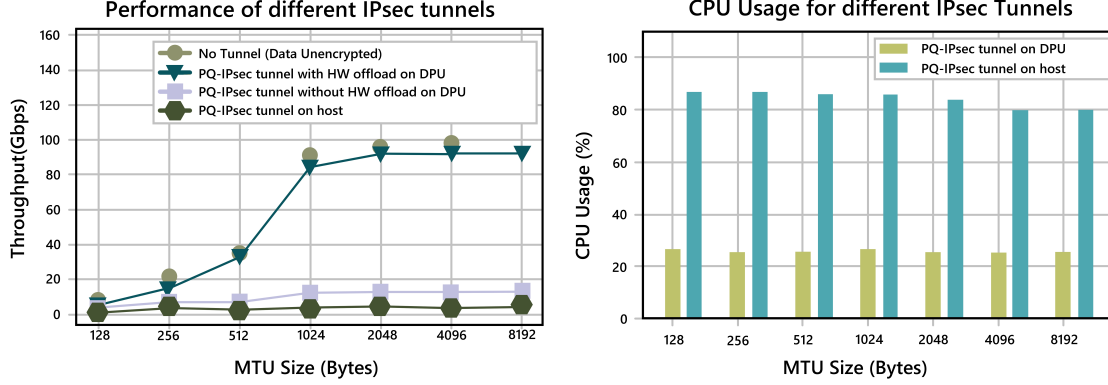


Figure 6.3: Throughput (left) and CPU utilization (right) for post-quantum IPsec tunnels in an end-to-end optical fiber transmission experiment. The evaluated configurations include unencrypted communication, server-based tunnel processing, and DPU-based processing with and without hardware acceleration. The results show that DPU hardware offloading maintains near line-rate throughput while significantly reducing host CPU utilization. Adapted from Figure 2 of [7], licensed under CC BY 4.0.

At scale, these experimentally observed processing costs accumulate across large numbers of secure connections and can directly reduce the capacity available for application workloads.

In large-scale environments, even modest per-operation overheads can accumulate across many simultaneous secure connections. This becomes especially critical in distributed workloads such as machine learning, where high volumes of internal communication must be secured efficiently. The architectural response introduced in Chapter 5 is evaluated here through DPU-assisted deployments, where cryptographic processing is distributed across specialized components and assessed in terms of throughput and host-resource utilization [12, 104].

System performance is further shaped by the choice of post-quantum algorithm, as different schemes exhibit trade-offs between computational cost, communication overhead, and implementation complexity. Each variant therefore reflects a different balance between security, size, and computational efficiency [6]. In the evaluated implementations, ML-DSA provides comparatively efficient signing and verification, which can make it attractive in throughput-oriented environments. However, this efficiency comes at the cost of significantly larger public keys and signatures compared to classical standards, which can negatively affect bandwidth-constrained deployments [10].

A further deployment-relevant distinction concerns the distribution of computational work between endpoints. In a client-server model, this distribution depends on which party signs and which party verifies. ML-DSA/ML-KEM provides a comparatively balanced

profile across common operations, while FN-DSA/ML-KEM offers compact signatures and efficient verification but more demanding key generation and signing. As a result, FN-DSA may benefit constrained clients when the client primarily verifies server signatures, but it may increase server-side cost when the server performs frequent signing operations. The practical advantage therefore depends on protocol role, authentication direction, bandwidth constraints, and endpoint capability [6].

Hash-based schemes such as SLH-DSA rely on well-established cryptographic assumptions, but exhibit significantly lower performance and very large signature sizes, in some cases reaching tens of kilobytes. This makes them difficult to use in latency-sensitive communication protocols, although they may remain relevant in environments with abundant bandwidth and computational resources where reliance on conservative hash-based assumptions is desirable [9, 79].

These differences illustrate that no single algorithm is universally optimal, and that selecting a post-quantum scheme requires careful consideration of the deployment context, including computational resources, bandwidth constraints, endpoint capabilities, and system requirements. These profiles suggest that ML-DSA/ML-KEM can be attractive where throughput is a primary constraint, whereas FN-DSA/ML-KEM may be attractive where signature size and verification cost dominate. Suitability remains dependent on workload, endpoint role, bandwidth, and implementation [3, 6].

In the very-high-throughput deployments examined here, hardware acceleration emerges as an important scalability mechanism for post-quantum-secured communication. By offloading computationally intensive operations, such as symmetric encryption and post-quantum handshakes, to specialized components like DPUs, systems can maintain line-rate performance while preserving host resources for application workloads [7]. This approach embeds cryptographic processing directly into the data path, allowing secure communication to operate efficiently even at high link speeds.

Experimental results show that hardware offloading can significantly increase effective system capacity while reducing CPU utilization, enabling secure communication at 100 Gbit/s and 200 Gbit/s line rates that are difficult to achieve with software-only implementations [7, 14]. Furthermore, emerging heterogeneous architectures enable parallel processing of post-quantum operations, which is particularly important in cloud environments where large numbers of secure connections must be supported concurrently [21].

Figure 6.4 illustrates this behavior by comparing throughput and CPU utilization under different deployment configurations in a 200 Gbit/s optical MACsec experiment. The three evaluated scenarios are unencrypted communication, software-based MACsec processing in the Linux kernel, and MACsec with cryptographic operations offloaded to a DPU. In this setting, the secure MACsec link is bootstrapped through quantum-secure key establishment using classical cryptography, PQC, and QKD, while data-plane protection is performed using AES-GCM [14].

The throughput results (left) show that while unencrypted communication reaches full link capacity, software-based encryption significantly limits performance. With Linux kernel MACsec, throughput remains around 300 Mbit/s for small MTU sizes, increases to approximately 5 Gbit/s with standard MTU sizes, and reaches only about 13 Gbit/s with jumbo frames. In contrast, when MACsec encryption is offloaded to the DPU, throughput increases to approximately 93.4 Gbit/s around standard MTU sizes and reaches 196 Gbit/s with jumbo frames, corresponding to more than 99% of the available channel capacity [14].

The CPU utilization results (right) further highlight the system-level impact. When MACsec encryption and frame processing are performed in software, the server-side CPU load reaches approximately 81% in the evaluated configuration. When the same cryptographic operations are offloaded to the DPU, CPU usage decreases to approximately 45%, freeing around 35% of host processing capacity. This demonstrates that cryptographic processing, when executed in software, not only limits throughput but also consumes a large portion of computational resources, whereas DPU offloading preserves both data-plane performance and host resources [14].

In the evaluated 200 Gbit/s MACsec configuration, DPU-based hardware offloading enabled encrypted throughput to approach the available link capacity while reducing host CPU load. The result therefore demonstrates the value of hardware-assisted execution for this class of very-high-throughput data-plane deployment rather than establishing hardware offloading as a universal requirement for all post-quantum systems.

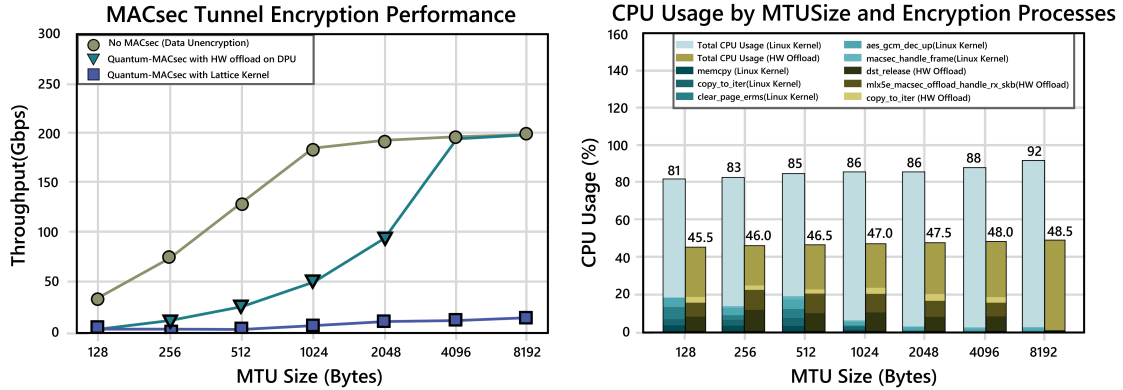


Figure 6.4: Throughput (left) and CPU utilization (right) for a 200 Gbit/s-regime quantum-secure MACsec optical link under different configurations: unencrypted transmission, software-based MACsec encryption in the Linux kernel, and hardware-accelerated MACsec encryption using DPUs. The results show that software-based processing reaches only a small fraction of channel capacity, while DPU offloading enables 196 Gbit/s encrypted throughput with jumbo frames and significantly reduces host CPU load. Adapted from Figure 2 of [14], with permission.

Overall, these observations demonstrate that performance and scalability are central to the practical deployment of PQC. The workload-dependent computational costs and increased data sizes associated with post-quantum schemes can introduce challenges at both the protocol and infrastructure levels. However, the experimental results also show that these costs can be managed when control-plane cryptographic complexity is isolated from high-throughput data-plane processing and when symmetric data protection is supported by hardware acceleration. The architectural mechanisms that enable this separation are examined in the following section.

6.4 Hardware Acceleration and Cryptographic Offloading

The performance measurements discussed in the previous section motivate a hardware-assisted execution model in which cryptographic processing is distributed across specialized hardware and software components. In high-throughput post-quantum systems, cryptographic processing can be distributed across specialized hardware and software components to reduce pressure on host processing, memory, and communication resources [9, 15].

To address these limitations, modern systems adopt an architectural approach known as cryptographic offloading. In this model, computationally intensive security operations are transferred from the host CPU to specialized hardware components, such as DPUs or SmartNICs. Unlike traditional network interface cards, which primarily handle packet transmission, these devices actively participate in the cryptographic lifecycle. They receive data from the host, perform encryption and authentication, and transmit secured packets without requiring continuous CPU involvement. This allows cryptographic processing to occur transparently from the perspective of applications, while reducing the burden on general-purpose processors.

The effectiveness of offloading depends not only on the raw speed of the accelerator, but also on its placement within the system. External accelerators may provide high cryptographic throughput, but their benefit can be reduced if data movement between the host, accelerator, and network interface introduces additional latency. DPUs address this limitation by placing programmable processing, cryptographic acceleration, and network interfaces within the same data-path-oriented platform. This locality makes them particularly suitable for line-rate post-quantum communication, where both cryptographic computation and packet transmission must be coordinated efficiently [9, 11].

A further distinction concerns the granularity of the offloaded operation. In IPsec, crypto offload accelerates the encryption and decryption operations, but leaves packet processing tasks such as ESP encapsulation and decapsulation in software. Packet offload, by contrast, moves both cryptographic operations and ESP packet processing into hardware. This distinction is important because the overhead of secure communication is not limited to the cipher itself. It also includes packet transformation, metadata handling, and interaction

with the network stack. Experimental results with PQC-enabled IPsec show that packet offload provides higher throughput than crypto-only offload, particularly for high-rate UDP traffic [18].

A representative example is an end-to-end post-quantum-protected DPU-to-DPU communication link in which post-quantum mechanisms are integrated into the control plane while symmetric encryption protects the data plane. In such deployments, ML-KEM can be used to establish post-quantum pre-shared key material, while post-quantum signature schemes such as ML-DSA, FN-DSA, or SLH-DSA support authentication. This post-quantum material is combined with classical key-establishment and authentication mechanisms during tunnel setup, after which AES-256 is used to encrypt application data at high throughput. This demonstrates that PQC can be integrated into a complete communication pipeline for high-capacity environments rather than evaluated only as isolated cryptographic primitives [3, 7, 9].

This also means that post-quantum algorithm selection cannot be separated from the network domain in which the algorithm is deployed. Different schemes impose different processing costs and produce cryptographic artifacts of different sizes, which affects their suitability for specific environments. Algorithms that are acceptable for intra-data-center or intra-rack communication may not be equally suitable for far-edge or resource-constrained deployments, where bandwidth, latency, and processing capacity are more limited. Consequently, the choice of post-quantum mechanism becomes a deployment-dependent architectural decision, shaped by both cryptographic security and the operational characteristics of the network segment [12].

However, cryptographic offloading should not be understood as an automatic decision to move all post-quantum operations away from the host. The suitability of offloading depends on the performance requirements of each function, including throughput, latency, computational intensity, memory pressure, power consumption, and integration complexity. It also depends on the capabilities of the available accelerator and on whether the performance gains justify the additional hardware, orchestration, and software-management costs. From this perspective, offloading is a system-design decision rather than a purely cryptographic optimization. Different post-quantum functions may require different execution locations depending on the workload, network segment, and deployment environment [12].

This execution model also supports so-called “zero-tax” architectures, in which infrastructure-related functions, including security processing, are offloaded from the host so that processing resources remain available for primary application workloads [12].

In this context, DPUs and SmartNICs evolve into active system components responsible for managing both networking and security operations, rather than serving as simple I/O devices [104]. The operating mode of such devices also affects the deployment model. In NIC mode, the device behaves primarily as a high-performance network interface with access to hardware offload capabilities. In DPU mode, the embedded processing

subsystem becomes active and can enforce traffic steering, policy control, and isolation through programmable components such as Open vSwitch (OVS). This creates a design trade-off. NIC mode can maximize raw throughput and simplify offload, while DPU mode provides stronger control over multi-tenant traffic and security enforcement [18]. These platforms integrate programmable processing units, typically based on energy-efficient ARM architectures, alongside dedicated hardware accelerators. Instruction-set extensions available in modern ARM architectures provide an additional layer of optimization. By exploiting built-in cryptographic and Single Instruction, Multiple Data (SIMD) capabilities, including NEON-based acceleration, implementations can improve selected post-quantum subroutines without relying exclusively on external accelerators [15].

Architecture-aware optimization is not limited to the ARM-based DPU platforms considered above. Intel reports that an AVX-512-optimized Keccak implementation in liboqs improves ML-KEM performance by up to approximately $1.54\times$ and ML-DSA by up to $1.64\times$ relative to its evaluated AVX2 implementation [120]. Independent RISC-V research reaches a similar conclusion from a different architectural direction. In the QUASAR hardware–software co-design, NTT and Keccak accounted for approximately 92–95% of ML-KEM and ML-DSA key-generation execution time on the target processor. Accelerating these kernels produced approximately $2.36\text{--}3.18\times$ end-to-end speedups across the evaluated NIST-standardized algorithm families with a 19% LUT increase over the base processor [85].

These results independently support the value of targeting dominant post-quantum computational kernels through architecture-specific acceleration. At the same time, the multi-layer measurements discussed in Section 6.3 show that primitive-level speedups should not automatically be interpreted as proportional reductions in protocol or application latency, since data movement, packet processing, and network conditions may become the dominant bottlenecks at higher layers [40].

The programmable cores execute control-plane logic, including authentication, identity verification, key establishment using post-quantum primitives, and retrieval of QKD-derived key material, while specialized hardware engines handle high-throughput data-plane operations such as symmetric encryption [5, 8]. This division is particularly important in triple-hybrid IPsec deployments, where software-based IKEv2 negotiation can combine classical, post-quantum, and QKD key material, while hardware-accelerated AES-256-GCM preserves near-line-rate packet protection on the data plane [8].

Figure 6.5 illustrates the architecture of the NVIDIA BlueField-2 DPU, highlighting the integration of embedded ARM cores, memory subsystems, hardware acceleration engines, and network interfaces within a single system-on-chip design.

In such platforms, embedded ARM cores can execute control-plane software, while integrated acceleration engines support cryptographic operations such as public-key processing and AES-GCM-based authenticated encryption.

This architecture enables the offloading of networking and cryptographic operations from the host CPU to the data path, supporting high-throughput and low-latency secure communication.

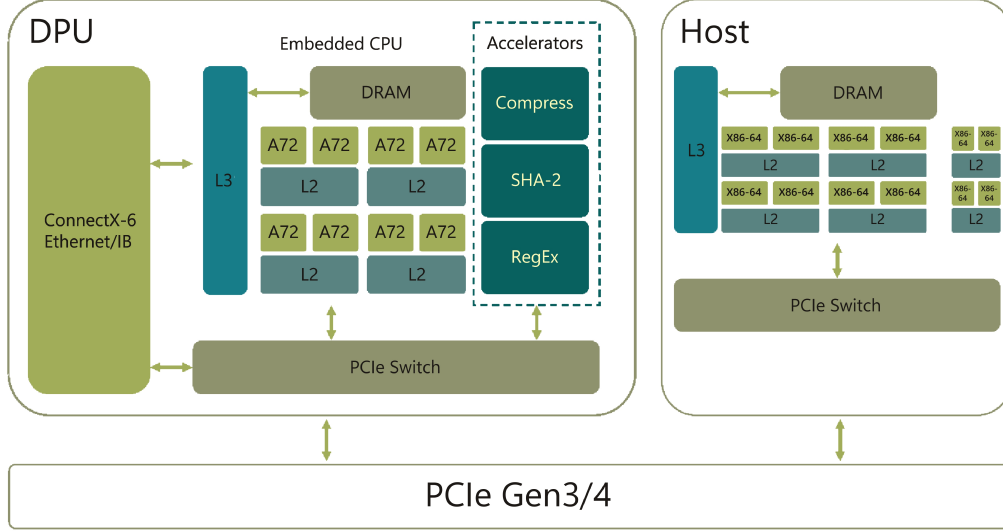


Figure 6.5: Architecture of the NVIDIA BlueField-2 DPU, integrating embedded ARM cores, memory subsystems, hardware accelerators, and high-speed network interfaces within a system-on-chip design. Adapted from Figure 4.2 of [1], with permission from the author.

These systems also support virtualization and traffic management through mechanisms such as SR-IOV and software-defined switching. This enables secure and isolated communication across multiple tenants without continuous reliance on host CPU involvement [18, 19]. In more advanced configurations, integrated accelerators enable parallel processing of large batches of cryptographic operations, further improving system efficiency [21]. This is particularly relevant for lattice-based schemes such as ML-KEM and ML-DSA, whose polynomial operations, including NTTs, can be vectorized and mapped effectively onto highly parallel GPU architectures. In such settings, DPUs no longer act only as network offload devices, but as heterogeneous cryptographic platforms that combine embedded CPU cores for orchestration with onboard GPU resources for massively parallel post-quantum computation [21].

This architecture maps crypto-agile negotiation and key establishment onto programmable execution paths while assigning continuous symmetric traffic protection to hardware acceleration. The resulting placement preserves software flexibility for evolving post-quantum mechanisms while sustaining line-rate encryption in the data path [3, 13].

The value of offloading is particularly visible in IPsec-protected data-center communication. When IPsec processing is executed entirely on host CPUs, encryption overhead can dominate the data path and substantially reduce effective link capacity; in high-speed settings, even a small number of IPsec connections may reduce a 100 Gbit/s link to below 10 Gbit/s.

Hardware acceleration on SmartNICs or DPUs addresses this bottleneck by offloading AES-256 encryption from the host and allowing encrypted traffic to be processed at line rate without consuming application-side CPU resources [20].

Recent DPU-based implementations further show that in-line MACsec can be supported at very high data rates by combining a software-defined control plane with hardware-offloaded data-plane encryption. In a 200 Gbit/s optical experiment, this approach enabled a quantum-resilient MACsec link to reach 196 Gbit/s encrypted throughput using commodity DPU/NIC offloads, demonstrating that line-rate link-layer protection is feasible when encryption is moved into the network data path [14].

A further architectural extension is the combined deployment of quantum-resilient IPsec and MACsec using different DPU generations according to protocol requirements. In one such design, BlueField-2 DPUs operating at 100 Gbit/s manage both the control plane and data plane for IPsec, establishing quantum-secure tunnels using post-quantum authentication and key exchange before forwarding encrypted IP traffic without host intervention. In parallel, BlueField-3 DPUs operating at 200 Gbit/s support MACsec, where the host performs crypto-agile control-plane operations, including hybrid key establishment, and transfers the resulting session key to the DPU for line-rate Ethernet-frame encryption. This illustrates how different offloading models can coexist within the same quantum-resilient infrastructure [16].

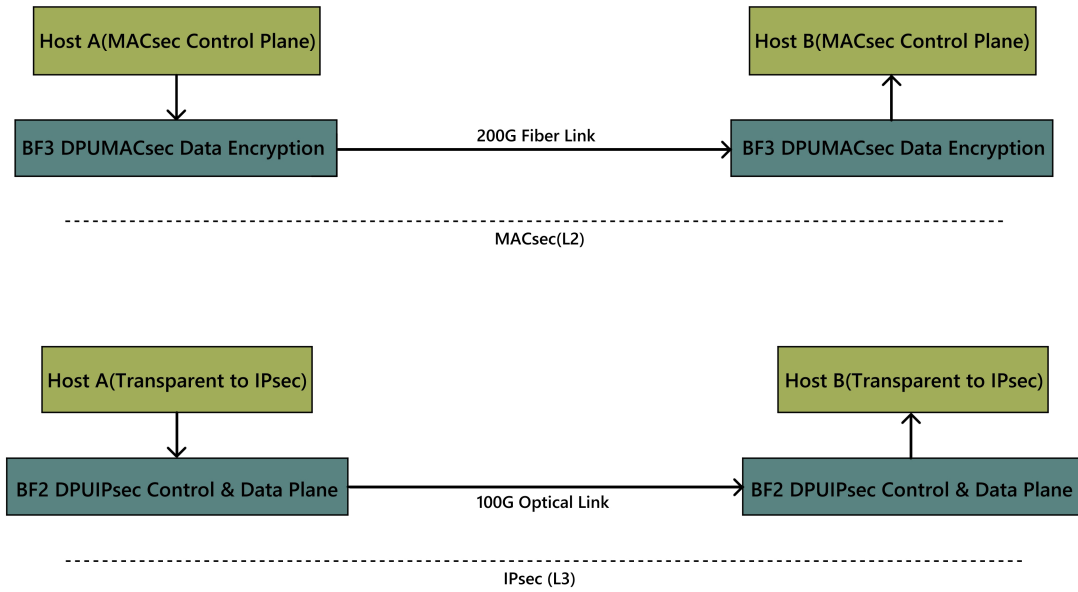


Figure 6.6: Combined architecture for quantum-resilient data transfer, showing the distribution of control-plane and data-plane operations across hosts and DPUs. Adapted from Figure 5 of [16], with permission.

Figure 6.6 summarizes this deployment model by showing how quantum-resilient data transfer can distribute control-plane and data-plane responsibilities across hosts and DPUs. The figure illustrates that different protocol layers can use different execution models.

IPsec can be managed more directly within the DPU, while MACsec can combine host-side crypto-agile key establishment with hardware-offloaded frame protection.

The introduction of hardware acceleration has a direct impact on system performance and scalability. By offloading cryptographic operations to specialized hardware, systems can maintain throughput levels close to their unencrypted baseline while significantly reducing the computational load on host processors. This enables infrastructures to support large numbers of concurrent secure connections without saturating available resources. In virtualized environments, hardware-assisted execution allows isolated network functions to operate efficiently, demonstrating that secure and scalable multi-tenant systems remain feasible even when post-quantum mechanisms are deployed. In the evaluated DPU-based systems, ARM-optimized implementations of the pre-standardization Dilithium and Kyber schemes reduced CPU-cycle requirements by factors ranging from 5 to 31 and from 8 to 57, respectively. Across related hardware-offloaded experiments, the achievable link capacity increased by more than a factor of 2.5 compared with the corresponding reference implementations. These results indicate that architecture-specific optimization and hardware offloading can be important scalability mechanisms for high-throughput PQC deployments [3, 7, 12, 15].

These experimental results provide concrete evidence for the architectural redistribution introduced in Chapter 5. In the evaluated systems, DPUs act as active components that manage data movement, security, and resource allocation while reducing the burden placed on host processors. This approach further enables so-called “zero-tax” environments, in which infrastructure-related tasks such as networking and security are offloaded from the host, thereby allowing processors to focus on application workloads [12]. In parallel, heterogeneous architectures combining different types of accelerators support large-scale parallel processing of cryptographic operations.

At the same time, multilayer security frameworks integrate protection across multiple protocol layers, for example by combining MACsec at the Ethernet layer with IPsec at the network layer. This provides protection across both local link-layer communication and routed IP-based communication, reducing reliance on any single security mechanism and limiting the impact of a compromise or misconfiguration at one layer [16]. These developments reflect a broader shift toward systems in which security is embedded directly into the data path rather than treated as an external function [13, 21].

Figure 6.7 illustrates this architectural transition in the context of modern data-center environments. The figure highlights both the large-scale interconnection of GPU clusters through spine-leaf topologies and the role of DPUs in enabling efficient post-quantum cryptographic processing through hardware-assisted offloading.

The need for acceleration also extends beyond the execution of post-quantum primitives themselves. In high-performance data-center and HPC environments, the overhead of quantum-resilient communication can appear in the surrounding communication pipeline,

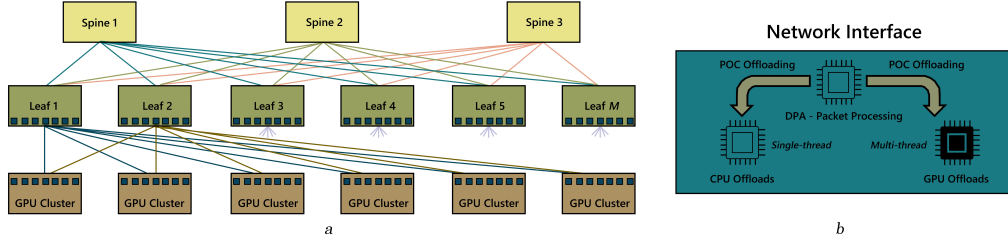


Figure 6.7: DPU-assisted architecture for post-quantum-secured data-center communication. (a) Spine-leaf data-center architecture interconnecting GPU processing clusters. (b) Heterogeneous DPU-based offloading model in which PQC-related operations are distributed across embedded CPU cores and onboard GPU accelerators, enabling batched post-quantum processing close to the network path. Adapted from Figure 1 of [21], licensed under CC BY 4.0.

including RDMA, remote procedure calls, message passing, key management, and the serialization or deserialization of cryptographic material. As the number of secure connections and key exchanges increases, these auxiliary operations can become part of the overall deployment cost of PQC. Offloading therefore concerns not only the acceleration of individual algorithms, but also the optimization of the network-control and data-movement mechanisms that support quantum-resilient communication at scale [12].

Overall, these observations, together with the architectural model illustrated in Figure 6.7, show that hardware acceleration is an important scalability mechanism for demanding, high-throughput PQC deployments. In such environments, distributing selected cryptographic and packet-processing tasks across specialized hardware and software components can reduce pressure on general-purpose processors and help sustain system performance as network rates and workloads increase. The importance of such acceleration therefore depends on the throughput requirements, workload characteristics, and architecture of the target deployment.

This execution model is summarized in Figure 6.8. The figure contrasts a traditional CPU-centric design with an emerging network-centric model in which programmable network interfaces and accelerators participate directly in security, packet processing, and data movement. In traditional architectures, the CPU acts as the central coordinator of computation, memory, storage, and communication. In the emerging model, selected responsibilities move closer to the data path through programmable network interfaces and hardware accelerators. This shift reduces host involvement in repetitive infrastructure tasks and allows cryptographic processing to scale with network throughput. In the context of PQC, the figure illustrates why deployment depends not only on algorithm choice, but also on where cryptographic operations are executed within the system.

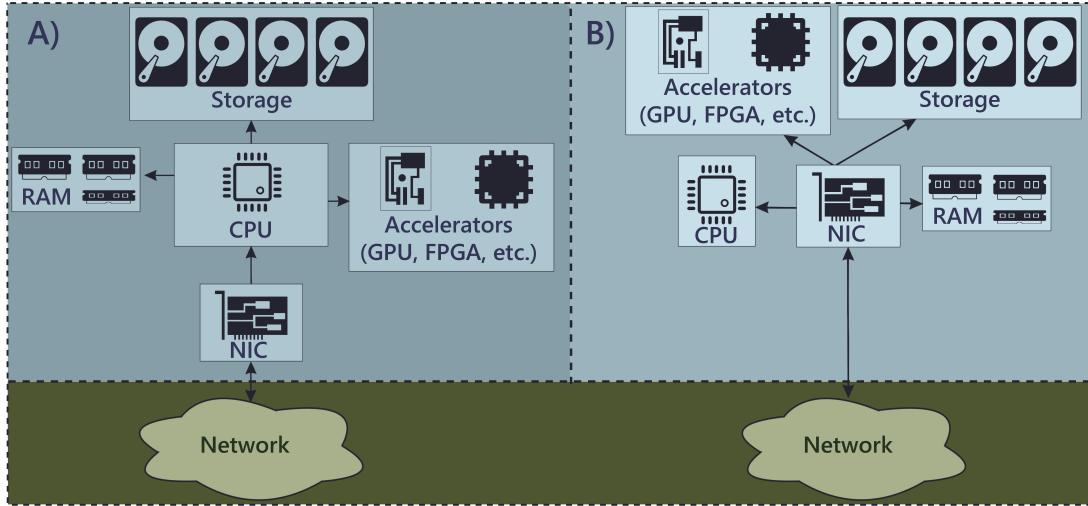


Figure 6.8: Evolution of the compute paradigm from CPU-centric architectures (A) to network-centric designs where SmartNICs and hardware accelerators orchestrate computation (B). Adapted from Figure 4.9 of [2], with permission from the author.

6.5 Post-Quantum Cryptography in Virtualized and Multi-Tenant Systems

The deployment of PQC in modern infrastructures extends beyond physical systems into virtualized environments. Within such settings, computing resources are abstracted and shared among multiple users. Quantum-resilient mechanisms must therefore be integrated into the software and networking stacks that govern virtual machines and containerized workloads. Virtual machines provide strong isolation by emulating independent systems with their own operating systems, making them suitable for environments where strict security boundaries are required. In contrast, containerized environments rely on a shared host operating system kernel, where isolation is achieved through mechanisms such as namespaces and control groups. While this approach improves efficiency and scalability, it introduces additional security considerations, as vulnerabilities at the kernel level may affect multiple workloads. In this context, network slicing enables the creation of multiple logical networks over shared physical infrastructure, allowing secure communication channels to be established between isolated workloads [18, 121, 122]. While SR-IOV provides high-performance access to network hardware within a single host, network slicing extends isolation across distributed infrastructures by leveraging mechanisms such as software-defined networking and network function virtualization [121, 122].

The need for such isolation is reinforced by the threat model of multi-tenant data centers. Although access from outside the data center is typically restricted by perimeter defenses, legitimate access to shared infrastructure may still be obtained through tenant-based service models. Once inside the infrastructure, an attacker may attempt to escape from an assigned virtual environment, escalate privileges, monopolize shared services, or observe

sensitive traffic generated by other workloads. In AI and high-performance computing environments, these risks extend to attacks against training pipelines, including data poisoning, model extraction, inference observation, and passive inspection of training data. As a result, post-quantum-secured communication in virtualized environments must be considered together with workload isolation, access control, and protection against internal data-center attack surfaces [20].

A further scalability challenge arises from the dynamic identity lifecycle of cloud-native and virtualized infrastructures. Modern data centers may need to generate and verify cryptographic identities for large numbers of containers, microservices, virtual machines, and AI workloads that are created and removed dynamically. In such environments, traditional identity provisioning mechanisms, including long-lived keys or conventional certificate issuance workflows, may become too slow or cumbersome to support high-churn deployments. This challenge becomes more complex when confidential computing and AI inference workloads require hardware-backed attestation and secure key provisioning before identities can be bound to specific workloads [21].

In real-world deployments, post-quantum cryptographic operations are integrated directly within these environments. The PQC handshake is executed within the virtualized guest, using algorithms such as ML-KEM for key establishment and ML-DSA for authentication. Once a quantum-secure key is established, high-throughput data protection based on symmetric encryption, such as AES-256-GCM, can be offloaded to specialized hardware components, including DPUs, thereby maintaining system performance. Experimental evidence indicates that such architectures can achieve near-native performance, reaching tens or even hundreds of gigabits per second depending on the underlying hardware platform [18].

This integration is facilitated by hardware-assisted virtualization technologies such as SR-IOV, a foundational mechanism for network slicing in modern data centers. SR-IOV allows a single physical network interface card to expose multiple virtual interfaces, known as VFs, which can be directly assigned to virtual machines [108, 109]. These VFs can be accessed through passthrough mechanisms such as Virtual Function I/O (VFIO), enabling direct memory access between the virtual machine and the network interface and bypassing the software layers of the hypervisor. This approach enables near-native I/O performance while maintaining strong isolation between tenants.

This interaction is illustrated in Figure 6.9, which shows how SR-IOV enables direct communication between virtual machines and the network interface through VFs. By allowing virtualized workloads to access NIC resources via Direct Memory Access (DMA), the mechanism reduces I/O overhead while preserving isolation between tenants.

Such mechanisms support hardware-assisted separation of virtual functions, allowing multiple tenants to share the physical interface while retaining traffic isolation in the evaluated configuration [18]. When operating in switchdev mode, the physical function of the net-

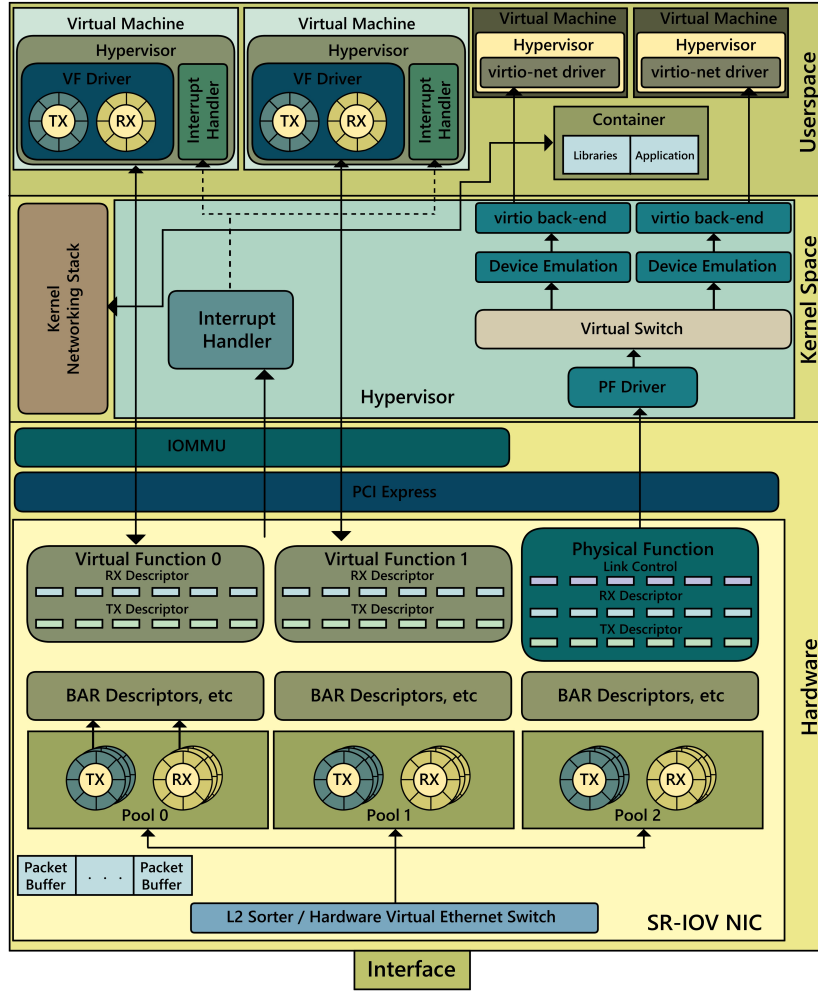


Figure 6.9: SR-IOV architecture. A physical network interface exposes multiple VFs that can be directly assigned to virtual machines, enabling near-native performance through direct memory access while maintaining isolation. Adapted from Figure 5.1 of [2], with permission from the author; original architecture based on [123].

work interface acts as a switching entity, offloading packet processing to hardware while embedded processing units coordinate traffic between VFs and the external network. This architecture allows secure communication to scale efficiently across virtualized environments while maintaining high performance and tenant isolation.

In multi-tenant systems, where multiple users share the same physical infrastructure, PQC plays a central role in maintaining both logical and cryptographic isolation. Each tenant operates within an isolated communication domain, often protected through dedicated encrypted tunnels. Under these conditions, post-quantum mechanisms ensure that sensitive data remains secure even in the presence of shared resources. Experimental evaluations show that, when combined with hardware-assisted virtualization and DPU-based packet steering, such environments can achieve performance levels approaching native operation. In a

100 Gbit/s BlueField-2 deployment, a virtualized PQC-IPsec network-slicing configuration with 16 virtual machines reached 71.79 Gbit/s, corresponding to 95.36% utilization of the baseline virtualized network performance [19]. In a later 400 Gbit/s BlueField-3 deployment, the same architectural principle scaled further: UDP-based PQC-IPsec traffic in a 16-VF-to-host configuration reached 392.98 Gbit/s, corresponding to 99.99% of the unencrypted baseline [18]. These results show that quantum-resilient network slicing can be practical at high data rates when virtualization, packet steering, transport behavior, and cryptographic processing are designed together.

In the evaluated high-throughput sliced environment, UDP made more effective use of parallel flows and hardware-assisted packet processing, whereas TCP remained more constrained by transport-layer behavior [18].

Encrypted tunneling can also reduce the exposure of sensitive backend services in shared environments. For example, RDMA services are performance-critical and may be vulnerable to monopolization or flooding attempts if malicious traffic reaches the service directly. When such traffic is encapsulated within an encrypted IPsec tunnel, packets that are not correctly protected can be discarded before reaching the RDMA endpoint. In this way, cryptographic protection contributes not only to confidentiality and integrity, but also to limiting the attack surface of high-performance communication services within multi-tenant data centers [20].

This deployment model is illustrated in Figure 6.10, which presents the architectural overview of a real experimental setup used to evaluate post-quantum secure communication in a virtualized environment. In this configuration, Server0 hosts multiple virtual machines that operate as independent client instances, each connected to the underlying network interface through dedicated VFs enabled by SR-IOV. These virtualized workloads interact with a DPU, which manages network traffic and supports cryptographic operations.

In this setup, each virtual machine is assigned a dedicated virtual interface, enabling isolated communication channels across shared hardware resources. The DPUs on both servers are directly interconnected through a high-speed 400 Gbit/s link, allowing the system to evaluate post-quantum secure communication under realistic high-throughput conditions without intermediate network devices. This architecture demonstrates how virtualization, hardware acceleration, and PQC can be combined to support scalable and isolated secure communication in modern data center environments.

Despite these capabilities, deploying PQC in shared environments introduces several challenges. Post-quantum schemes impose higher computational effort and involve larger data representations, which place additional demands on processing and communication resources. In particular, the size of cryptographic artifacts can strain bandwidth and affect protocol behavior, while the computational overhead further increases existing infrastructure pressure. These challenges are further intensified by the need to adapt existing protocols and software stacks to support new key exchange mechanisms and hybrid

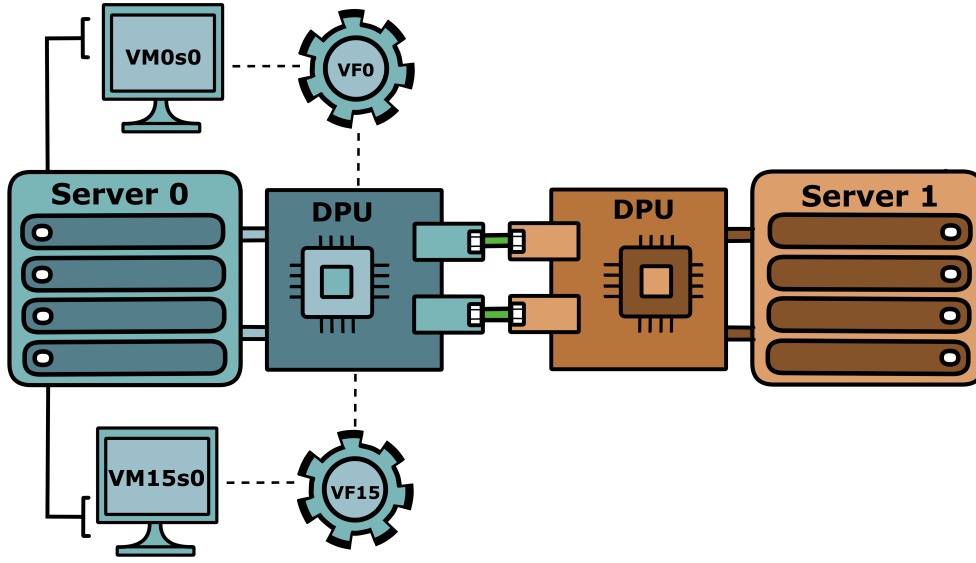


Figure 6.10: Architectural overview of the experimental setup for post-quantum secure communication in a virtualized environment. Multiple Virtual Machines (VMs) on Server0 are connected to VFs exposed by the DPU via SR-IOV. These communicate with a remote server (Server1) through a direct 400 Gbit/s DPU-to-DPU link. Original illustration based on the experimental setup reported in [18].

cryptographic configurations [9, 18, 19]. In addition, post-quantum deployments must remain adaptable as standardization and cryptanalytic assessments continue to evolve [84].

In operational settings, these challenges are addressed using the hardware-assisted execution model introduced in Chapter 5 and examined in Section 6.4. Within virtualized environments, this model allows post-quantum control-plane functions and high-throughput data protection to be distributed according to the capabilities of the available processing components. Post-quantum algorithms such as ML-KEM and ML-DSA are primarily used during the control phase to perform authentication and establish shared secrets, typically within software components, often executed on energy-efficient ARM cores within DPUs [3]. Protocols are adapted to support these mechanisms, for example through extensions such as RFC 9370, which enable multiple successive key exchanges within IKEv2 to accommodate post-quantum and hybrid constructions [13, 102].

Once established, the resulting session keys are used by hardware-accelerated mechanisms to protect high-throughput data flows, typically relying on symmetric encryption schemes such as AES-256-GCM. Demonstrated implementations show that such architectures can support high-speed secure communication, including PQC-enabled IPsec tunnels at approximately 100 Gbit/s in earlier configurations and scaling to several hundreds of gigabits per second, including 200 Gbit/s MACsec links and 400 Gbit/s DPU-based deployments, while preserving system scalability [7, 13, 14, 18]. These results also support multilayer protection models in which IPsec secures routed IP traffic while MACsec secures Ethernet-frame transmission within the local or data-center fabric [16]. In more advanced configura-

tions, hybrid constructions combine classical cryptography, post-quantum mechanisms, and QKD-derived key material to diversify the sources of session-key material during the transition. This use of QKD should be understood as contributing to key distribution under appropriate implementation and authentication assumptions, rather than as a replacement for post-quantum authentication or digital signatures [13].

The selection of post-quantum cryptographic mechanisms in shared infrastructures is inherently context-dependent and shaped by factors such as computational resources, network capacity, power constraints, and endpoint roles [6, 17]. In high-performance data centers, where bandwidth is abundant and throughput is critical, schemes with efficient signing and verification profiles, such as ML-DSA, may be attractive for large numbers of server-to-server authentication events. In bandwidth-constrained or client-constrained settings, compact signatures and efficient verification can make FN-DSA attractive when the constrained endpoint primarily verifies signatures. However, this advantage depends on the authentication direction, because FN-DSA signing and key generation are more demanding. These differences highlight that the deployment of PQC is shaped not only by algorithmic properties, but also by the operational role of each endpoint and the constraints of the target environment.

A key requirement in such environments is the ability to parallelize cryptographic operations across multiple virtual machines and network interfaces, allowing secure communication to scale with the number of tenants and workloads sharing the same physical infrastructure [18, 19]. Virtualization and multi-tenancy introduce deployment challenges that extend beyond algorithm and protocol design. In shared infrastructures, systems must also ensure scalability, isolation, and efficient use of resources across multiple workloads. As a result, PQC becomes part of a broader system design problem, where security mechanisms must operate efficiently within complex and distributed environments.

Chapter 7

Discussion and Practical Implications

This chapter examines the broader implications of deploying PQC within modern computing and communication systems. Building on the algorithmic foundations presented in earlier chapters and the system-level analysis developed in Chapters 5 and 6, the focus shifts toward synthesizing the key observations that arise when these mechanisms are evaluated in real-world environments. It first revisits the transition from algorithmic security to system-level deployment, highlighting how performance constraints, protocol integration, and architectural design influence the behavior of quantum-resistant schemes. The discussion then outlines the main lessons derived from infrastructure-level implementations, emphasizing recurring design patterns and limitations observed across different deployment settings. The chapter subsequently considers the practical implications for secure computing infrastructures, examining how system architectures, communication protocols, and performance engineering must adapt to support post-quantum mechanisms at scale. Finally, it discusses emerging directions for quantum-resilient architectures, situates the networking-focused analysis within the broader post-quantum migration landscape, and concludes by reflecting on the role of PQC as an integral component of modern digital systems.

7.1 From Algorithmic Security to System-Level Deployment

The development of PQC has traditionally been driven by the need to construct cryptographic primitives that remain secure against both classical and quantum adversaries. In this context, theoretical approaches focus on the design of algorithms grounded in mathematically hard problems, such as lattice-based, hash-based, or code-based constructions, and on establishing formal security guarantees against attacks enabled by quantum algorithms such as Shor's or Grover's. These analyses typically treat cryptographic mechanisms

in isolation, abstracting away the constraints imposed by real-world systems. In applied settings, however, these mechanisms operate within complete computing and communication infrastructures, where they must operate within software stacks, protocol frameworks, and hardware platforms under practical performance constraints.

From this perspective, the evaluation of cryptographic schemes extends beyond correctness and security proofs to encompass system-level behavior. In applied settings, the suitability of a post-quantum algorithm is determined not only by its resistance to attacks, but also by its impact on throughput, latency, and resource utilization in real deployments. This includes factors such as hardware-specific optimizations, integration into protocols like TLS, IPsec, and MACsec, and the ability to sustain high-speed communication at data center line rates. The analysis developed in the previous chapters shows that this transition introduces a fundamental shift. A scheme that is secure in theory may still be impractical if it cannot be supported efficiently by the underlying infrastructure.

This shift also reveals a broader transformation in the role of cryptography within modern systems. Cryptographic mechanisms are no longer confined to isolated components, but are embedded throughout the entire data path of digital infrastructure. They operate across multiple layers of the networking stack, from link-layer protection to transport-level security, and are involved in every stage of data processing and transmission. Consequently, modifications to cryptographic primitives propagate across protocols, packet structures, and system architectures, affecting not only security but also communication efficiency and overall system behavior.

This observation also revisits one of the central assumptions of classical cryptography discussed in Chapter 2. Classical secure infrastructures were able to scale because their underlying computational assumptions remained stable for decades. Public-key mechanisms provided authentication and key establishment, while symmetric cryptography protected bulk data efficiently in the data plane. This division allowed cryptographic mechanisms to become deeply embedded in protocols, operating systems, hardware accelerators, and trust infrastructures. The post-quantum transition disrupts this stability. It does not merely require stronger parameters, but changes the assumptions around which deployed systems were originally designed.

As established in Chapters 2 and 3, post-quantum migration primarily changes the public-key mechanisms used for authentication, key establishment, certificate validation, and key management, whereas bulk symmetric protection remains structurally usable with appropriate security parameters.

The resulting picture is that PQC must be understood as a system-level discipline. Its effectiveness depends not only on theoretical security guarantees, but also on its ability to operate within the constraints of modern infrastructures, where performance, scalability, interoperability, implementation security, and architectural integration all play a role.

7.2 Lessons from Infrastructure-Level and Applied Post-Quantum Cryptography

The transition of PQC from theoretical design to real-world deployment reveals a clear gap between algorithmic security and operational feasibility. While theoretical work focuses on mathematical hardness and resistance to quantum attacks, real systems must operate under constraints related to processing capacity, memory usage, and communication efficiency. Cryptographic mechanisms cannot be evaluated in isolation, since their behavior is tightly coupled with the infrastructure in which they are deployed. In the high-throughput systems examined here, processing can become the limiting resource even when substantial link bandwidth remains available, while larger post-quantum artifacts increase pressure on communication resources. Even relatively small changes at the cryptographic level can therefore propagate across the entire system, requiring coordinated adaptation rather than simple substitution of primitives.

The applied results in Chapter 6 show that increased complexity during post-quantum or hybrid session establishment does not necessarily translate into equivalent degradation of sustained traffic performance. In the evaluated triple-hybrid IPsec deployment, classical, post-quantum, and QKD-derived key material increased control-plane complexity, while hardware-offloaded symmetric protection allowed the data path to remain close to line rate [8]. This demonstrates the practical value of isolating expensive session-establishment operations from continuous traffic processing.

A further lesson is that post-quantum deployment must be evaluated across complete communication paths rather than isolated cryptographic operations. In virtualized network-slicing environments, performance depends on the interaction between PQC handshakes, IPsec encapsulation, transport protocols, SR-IOV-based isolation, VF assignment, and DPU-based packet steering. Experimental results show that quantum-resilient network slicing can achieve high throughput when these components are designed together, but they also reveal that protocol choice remains decisive. UDP can scale close to baseline performance across many VFs, whereas TCP remains more strongly limited by transport-layer overhead and encapsulation effects [18, 19].

Although the exact throughput and latency values are configuration-dependent, these results consistently demonstrate that deployment architecture strongly influences whether post-quantum security can be achieved without unacceptable performance loss.

For the very-high-throughput IPsec and MACsec environments evaluated in Chapters 5 and 6, hardware offloading emerges as an important architectural mechanism for sustaining near-line-rate protected traffic while preserving host resources. This should not be interpreted as a universal requirement for post-quantum deployment. Independent measurements show that software implementations of ML-KEM and ML-DSA can be

practical in less throughput-intensive server, client, and embedded environments, where different bottlenecks may dominate [40, 91, 92].

At a larger scale, these observations highlight that performance becomes one of the dominant practical constraints. The additional computational and communication overhead introduced by PQC amplifies existing infrastructure costs, particularly in environments where large numbers of secure connections must be maintained simultaneously. As a result, scalability remains a central challenge. Even when individual operations appear efficient, their cumulative cost across millions of connections can lead to significant system-level limitations.

Another important lesson concerns the timing of migration. As discussed in Section 3.4, the HNDL threat creates a mismatch between data lifetime and cryptographic lifetime. In operational terms, this is compounded by migration time, including the time required to update deployed infrastructures. Consequently, PQC deployment must be treated as a present infrastructure-planning problem rather than as a future reaction to quantum hardware availability.

The standardization of post-quantum algorithms provides an essential foundation for this transition, but it does not eliminate the need for careful system-level evaluation. Standardized mechanisms such as ML-KEM, ML-DSA, and SLH-DSA provide deployable building blocks, yet their behavior differs significantly across workloads and environments. Algorithm diversity remains important because different post-quantum families offer different balances between performance, artifact size, implementation complexity, and cryptanalytic maturity. The practical lesson is therefore not that a single standardized primitive solves the migration problem, but that standardized primitives must be selected, combined, and deployed according to infrastructure constraints.

Independent production deployments and academic measurements support the same broader conclusion from different system environments. Standardized ML-KEM key establishment is often computationally manageable on contemporary hardware, while artifact size, post-quantum authentication, packetization, protocol processing, interoperability, and network conditions can become the dominant deployment constraints [40, 90, 96, 117]. At the same time, the deployment of hybrid ML-KEM across cloud, browser, mobile, CDN, and VPN infrastructures shows that the practical challenge is increasingly shifting from basic algorithm availability toward infrastructure-wide migration and cryptographic agility [94, 95, 97, 112].

These findings show that effective deployment of PQC depends on a combination of algorithmic design and architectural support. Security can no longer be considered independently of system performance, and successful implementations require close coordination between cryptographic mechanisms, communication protocols, and hardware infrastructure.

7.3 Practical Implications for Secure Computing Infrastructures

A first major implication concerns where cryptographic processing is executed within the system. In the very-high-throughput environments considered here, cryptographic and packet-processing demands can place substantial pressure on general-purpose host processors operating alongside application workloads. This creates a strong incentive to redistribute selected functions toward components located closer to the data path, such as DPUs and SmartNICs. Hardware-assisted execution can therefore preserve host resources while supporting high-throughput quantum-resilient communication. The practical significance of this architectural redistribution is demonstrated by the experimental deployments discussed below.

The demonstration of near line-rate post-quantum-secured optical communication using DPUs illustrates that this architectural shift is not only conceptual, but already practical in high-speed inter-data-center scenarios [7].

Similarly, the demonstration of quantum-secure MACsec at 196 Gbit/s shows that link-layer protection can also be moved into the accelerated network data path, enabling high-capacity intra- and inter-data-center communication without imposing the full cryptographic cost on host processors [14].

A second implication arises at the level of communication protocols, which must be adapted to accommodate the structural properties of PQC. The larger size of public keys, ciphertexts, and signatures affects how protocol messages are constructed and transmitted, often requiring support for increased payload sizes and more efficient handling of fragmentation. At the same time, classical key exchange mechanisms are replaced or complemented by post-quantum constructions, such as key encapsulation mechanisms, which alter handshake behavior and introduce additional processing steps. As discussed in Chapter 6, these changes typically do not lead to the replacement of existing protocols, but rather to their extension, particularly in the control plane. Hybrid approaches combining classical and post-quantum mechanisms provide an important transition mechanism in this context, supporting interoperability while diversifying cryptographic assumptions.

These protocol-level changes also affect trust infrastructures. Since PKI relies on digital signatures, certificate chains, and long-lived trust anchors, post-quantum migration must address certificate formats, certificate-chain validation, signature verification cost, and certificate lifecycle management. Larger post-quantum public keys and signatures can increase certificate sizes and handshake payloads, while hybrid certificates may further expand the amount of cryptographic material that must be transmitted and verified. As a result, PKI migration is not only a matter of replacing signature algorithms; it also requires operational planning across certificate authorities, software libraries, validation logic, and deployed endpoints [93, 117].

DPU-based IPsec experiments using post-quantum authentication, ML-KEM-based key establishment, and key-mixing procedures show that this transition can be implemented without a complete redesign of the communication stack. In these deployments, post-quantum key material is established externally to the native IPsec negotiation and then installed through the pre-shared key method, allowing the IPsec data plane to continue using AES-256-GCM for high-throughput packet protection [9, 10].

These experiments also show that post-quantum-secured infrastructure must be evaluated across different traffic patterns. In an east–west intra-data-center scenario, DPU-offloaded IPsec can sustain 100 Gbit/s line-rate encrypted communication, demonstrating feasibility for high-throughput internal data-center traffic. In a north–south scenario, a wireless edge client can establish a post-quantum-secured IPsec tunnel to a cloud-side DPU, reaching approximately 0.486 Gbit/s. This distinction is important because internal data-center traffic and external client-to-cloud traffic impose different constraints on bandwidth, latency, endpoint capability, and cryptographic workload placement [10, 17].

From a performance engineering perspective, the deployment of PQC introduces a set of constraints that must be carefully managed. The computational complexity of post-quantum algorithms, combined with the increased size of cryptographic artifacts, directly affects both processing efficiency and communication overhead. Operations such as hashing and polynomial arithmetic can account for a substantial portion of execution time in software implementations, while larger data structures increase bandwidth consumption and influence packet-level behavior. As a result, performance optimization becomes closely tied to both architectural design and implementation strategy. Hardware-assisted execution and specialized instruction sets can mitigate these effects in high-throughput deployments [7, 15, 85].

Packet-level configuration is also a practical performance variable. In IPsec deployments, throughput depends not only on cryptographic offloading, but also on parameters such as the MTU. Experimental results show that small packet sizes reduce effective throughput, whereas larger MTU values allow DPU-offloaded IPsec traffic to converge toward 100 Gbit/s line rate. This indicates that post-quantum-secured communication must be engineered at both the cryptographic and packet-processing levels [10].

A further practical implication is that secure networking infrastructure can contribute to operational security and intrusion detection. In encrypted data-center environments, the network does not only transport protected traffic; it can also provide telemetry about abnormal behavior. For instance, a high number of incorrectly protected packets destined for sensitive services such as RDMA endpoints may indicate attempted abuse, misconfiguration, or denial-of-service activity. Monitoring such events allows the infrastructure to support anomaly detection and incident response, complementing traditional perimeter defenses. This is particularly relevant in multi-tenant and AI-oriented data centers, where attackers

may obtain legitimate access to shared resources and where malicious behavior must be detected inside the infrastructure itself [20].

These implications extend directly to cloud and virtualized environments, where secure multi-tenant operation depends on both logical and cryptographic isolation. In such settings, post-quantum mechanisms must be integrated into virtualization layers and networking frameworks, ensuring that secure communication can be maintained across shared physical infrastructure. Technologies such as SR-IOV allow virtual machines to access network resources with near-native performance while preserving isolation between tenants. Experimental PQC-enabled network-slicing deployments show that this model can support high-throughput secure communication when each virtual machine is assigned a dedicated VF and packet steering is coordinated through DPU-based mechanisms such as OVS. These results imply that quantum-resilient multi-tenant infrastructure is practical, but only when virtualization, tunnel configuration, transport protocol behavior, and hardware offloading are treated as part of the same system design problem [18, 19]. Nevertheless, maintaining scalability across large numbers of secure connections remains a challenge, particularly in environments where resources must be shared dynamically among multiple workloads.

In practical deployment scenarios, the transition to PQC requires gradual and adaptable strategies. Rather than replacing classical mechanisms abruptly, real-world systems rely on hybrid approaches that combine classical and post-quantum techniques within the same protocol framework. Hybrid approaches can diversify cryptographic dependencies during migration, provided that their component mechanisms are combined and authenticated appropriately. At the same time, the concept of crypto-agility becomes increasingly important, as systems must be capable of adapting cryptographic mechanisms over time without requiring fundamental redesign. These considerations demonstrate that the integration of PQC is not a single-step process, but an ongoing evolution that must balance security, performance, and adaptability.

Crypto-agility should be understood as a long-term architectural property rather than only as a migration mechanism. Post-quantum algorithms will continue to evolve, implementation guidance may change, and new cryptanalytic results may alter confidence in particular schemes. Systems must therefore support algorithm negotiation, hybrid operation, parameter updates, certificate replacement, and controlled deprecation of vulnerable mechanisms. Without such flexibility, infrastructures risk repeating the same rigidity that makes the current transition difficult [32, 33].

This adaptability must also extend to key renewal and long-lived security-association management. If rekeying mechanisms fall back to classical key exchange, then an initially post-quantum-protected tunnel may lose its intended quantum-resilient security properties during renewal. Practical deployments therefore need rekeying procedures that preserve security properties equivalent to those intended during initial session establishment. Until

native post-quantum rekeying is fully supported, one deployment strategy is to rerun the post-quantum authentication and key-establishment procedure whenever fresh key material is required [10].

A further implication concerns the role of QKD within hybrid infrastructures. QKD can contribute independent key material in network segments where suitable quantum links, key-management infrastructure, and physically secure access to QKD equipment are available. Its role should be understood as supporting key distribution under appropriate implementation and authentication assumptions, not as replacing post-quantum authentication or digital signatures. As a result, practical infrastructures are likely to combine QKD-assisted hybrid mechanisms in specialized high-security links with classical–post-quantum hybrid mechanisms in segments where quantum links are not feasible. This reinforces the need for crypto-agile protocol designs that can move between single, hybrid, and triple-hybrid configurations depending on deployment constraints [13].

A related implication is that quantum-resilient infrastructures are likely to adopt layered protection rather than relying on a single secure protocol. Combining mechanisms such as MACsec and IPsec allows organizations to protect traffic at different layers of the communication stack, supporting both local data-center fabrics and routed network paths. This layered model reduces dependence on any single protocol boundary and provides a practical basis for scalable quantum-resilient data transfer [16].

7.4 Future Directions for Quantum-Resilient Architectures

Despite the progress achieved in recent years, the deployment of PQC still faces several practical limitations. As highlighted in Chapters 5 and 6, software performance can become a significant challenge in high-throughput environments, particularly for lattice-based schemes that rely on operations such as Keccak-based hashing and polynomial arithmetic. In optimized implementations, these computational kernels can account for a substantial portion of execution time, creating strong incentives for vectorization, architecture-specific optimization, and hardware assistance when very high throughput is required [85, 120]. In addition, connection-oriented protocols can experience noticeable performance degradation due to the increased overhead associated with post-quantum handshakes [18, 40]. These observations indicate that the transition from research prototypes to production-ready systems is still ongoing, and that further optimization is required to support efficient large-scale deployment.

These limitations are likely to increase the importance of hardware-assisted execution in high-throughput environments. As network rates and post-quantum workloads continue to grow, selected security, packet-processing, and data-movement functions are likely to be integrated more closely with programmable network interfaces and specialized accelerators. Future work therefore concerns not only faster cryptographic primitives, but also how these

functions should be distributed across heterogeneous computing platforms while preserving flexibility, interoperability, and efficient resource use.

From a protocol perspective, these architectural changes are reflected in the increasing importance of flexibility and adaptability. Current developments indicate a growing reliance on hybrid constructions that combine classical cryptography, post-quantum schemes, and, in some cases, QKD. Such constructions can diversify cryptographic dependencies when the component mechanisms are combined using an appropriate protocol design. In parallel, protocols are evolving to support crypto-agility, enabling systems to adapt to new cryptographic standards as they emerge. Another important direction is the adoption of multi-layer security models, where protection is applied at multiple levels of the networking stack, reducing the risk of a single point of failure.

Hardware acceleration is expected to play an increasingly important role in this evolution, particularly in environments where cryptographic processing must coexist with very-high-throughput networking and other compute-intensive workloads. As data rates continue to grow, specialized hardware can reduce host-resource consumption and accelerate dominant operations such as polynomial arithmetic, the NTT, and Keccak-based hashing. Independent results across x86 vectorized implementations and RISC-V hardware-software co-designs likewise show substantial gains from architecture-specific acceleration of these computational kernels [85, 120]. At the same time, software implementations remain practical for many server, client, and embedded workloads, indicating that the need for hardware acceleration depends on the target throughput, algorithm, workload, and system architecture. Hardware support should therefore be understood as an important scalability mechanism for demanding deployments rather than as a universal prerequisite for post-quantum security.

Looking ahead, deployment models are likely to move toward more integrated and adaptive security infrastructures. One important direction is the development of systems that support dynamic updates of cryptographic mechanisms, enabling long-term adaptability without major architectural changes. Hybrid deployment strategies will continue to play a key role during the transition period, balancing security and compatibility. At a broader level, security mechanisms are expected to extend across the entire data lifecycle, protecting data in transit, at rest, and during processing. These trends indicate a shift toward making post-quantum security an intrinsic property of system design rather than an external feature.

This direction is particularly important for confidential AI and large-scale training workloads. As AI models grow and training data becomes increasingly sensitive, data-center fabrics must protect data not only while it is transmitted between nodes, but also while it is stored and processed. Confidential computing extends this protection model by combining encryption for data at rest and in transit with protected execution environments for data in use. In this broader architecture, PQC contributes to quantum-resilient communication and

key establishment, while trusted execution and privacy-preserving computation techniques address the protection of data during processing. Future quantum-resilient infrastructures are therefore likely to combine post-quantum-secured networking, hardware-assisted encryption, trusted execution environments, and operational monitoring into a unified security model for confidential high-performance computing and AI workloads [20].

Several research challenges remain open. The gap between theoretical security guarantees and practical implementations requires further investigation, particularly for optimized versions of post-quantum algorithms. Improvements in the efficiency of core operations, such as hashing and polynomial arithmetic, are necessary to reduce computational overhead. In addition, hybrid cryptographic constructions require deeper analysis when deployed within complex protocol environments. Finally, scalability in highly dynamic systems, such as multi-tenant cloud infrastructures, remains a key challenge, as these environments must support large numbers of secure connections without compromising performance. Addressing these issues will be essential for the long-term adoption of PQC in real-world systems.

Implementation security is another important direction. The practical security of post-quantum schemes depends not only on the hardness of their underlying mathematical problems, but also on secure implementation. Constant-time execution, side-channel resistance, reliable randomness, secure key handling, and correct integration into protocol logic are essential for preventing attacks that bypass the formal security model. This is particularly relevant for optimized implementations, where performance improvements must not introduce timing leakage, memory-access leakage, or other implementation-level vulnerabilities [87, 88, 89].

7.5 Broader Post-Quantum Migration Landscape

The networking protocols emphasized in this book represent only part of the broader post-quantum migration landscape. Similar transition pressures are emerging in naming infrastructure, remote administration, secure messaging, software distribution, and device roots of trust. Recent IETF engineering guidance emphasizes that post-quantum migration may require adaptation of protocols and infrastructure rather than a one-for-one replacement of cryptographic algorithms, while current European standardization analysis shows parallel migration activity across protocol families including SSH and CMS/S/MIME [93, 124]. These developments reinforce that the system-level constraints examined for TLS, IPsec, and MACsec recur in other parts of digital infrastructure.

Domain Name System Security Extensions (DNSSEC) provides a particularly direct example of the communication impact of post-quantum signatures. Keys and signatures are carried in DNS records and responses, so the larger artifacts associated with post-quantum signature schemes can substantially increase response sizes. Current IETF work is therefore

examining mechanisms that contain this overhead by separating where larger post-quantum algorithms are used within the DNSSEC signing hierarchy rather than applying the same cost uniformly to every signed response [125]. This remains work in progress, but it illustrates the same relationship between cryptographic artifact size, protocol structure, and network behavior observed elsewhere in this book. In contrast, SSH has already reached operational deployment of hybrid post-quantum key establishment. OpenSSH 10.0 made the hybrid ML-KEM-768/X25519 mechanism `mlkem768x25519-sha256` the default for key agreement [126]. The differing maturity of DNSSEC and SSH illustrates why migration must be evaluated separately for each protocol ecosystem.

Secure messaging presents a related transition at the certificate and message-format level. S/MIME uses the CMS framework for signed and encrypted message structures. Recent IETF standards now specify the use of ML-DSA signatures in CMS and the use of ML-KEM for CMS recipient key establishment [127, 128, 129]. These standards provide core post-quantum building blocks for secure messaging, although practical migration still depends on certificate lifecycle management, client support, interoperability, and deployment policy. This reflects the broader PKI challenge discussed earlier in the book because standardizing a primitive is only one part of replacing the trust infrastructure that depends on it.

Code signing and firmware authentication expose an additional lifecycle constraint because signed artifacts and verification roots may remain operational for many years. NIST SP 800-208 identifies long-lived deployments that are difficult to modify after deployment, including firmware-update authentication for constrained devices, as an intended application area for stateful hash-based signatures such as LMS and XMSS [83]. For U.S. National Security Systems, CNSA 2.0 similarly identifies LMS and XMSS for specialized software- and firmware-signing applications [32]. These use cases differ from short-lived session establishment because the verification mechanism itself may be embedded into long-lived hardware or secure-boot infrastructure. Post-quantum migration may therefore need to reach roots of trust and update mechanisms earlier than other parts of the system.

These domains show that the post-quantum transition extends well beyond secure network tunnels while preserving the same underlying system-level concerns. Larger cryptographic artifacts interact with protocol formats and transport constraints; new algorithms require interoperability and certificate or key-management changes; and long-lived systems require cryptographic agility before their existing roots of trust become difficult to replace. The networking and hardware-focused analysis developed in this book should therefore be understood as one part of a broader infrastructure-wide transition toward quantum-resilient security.

7.6 Concluding Remarks

Overall, the analysis presented in this book shows that the transition to PQC extends beyond the development of new cryptographic algorithms. Quantum-resilient security requires changes across communication protocols, trust infrastructures, implementation practices, hardware-assisted execution paths, virtualization layers, and operational processes.

A central insight is that post-quantum migration affects different parts of the system in different ways. Key establishment, authentication, certificate validation, and key management require new public-key mechanisms, while high-throughput data protection can continue to rely on symmetric encryption with suitable security margins. This division explains why migration is both a cryptographic and an architectural problem.

Ultimately, PQC represents a transformation in how security is realized in modern systems. Its long-term success will depend not only on the strength of standardized algorithms, but also on the ability of infrastructures to deploy, accelerate, update, monitor, and operate these mechanisms over time. Quantum-resilient security must therefore be understood as a continuing system-design challenge rather than a one-time cryptographic replacement.

List of Acronyms

AEAD Authenticated Encryption with Associated Data.

AES Advanced Encryption Standard.

AI Artificial Intelligence.

ANSSI French National Cybersecurity Agency.

ARM Advanced RISC Machine.

ASCON Authenticated Security Combined with Nonce.

BHT Brassard–Høyer–Tapp.

BSI Federal Office for Information Security.

CA Certificate Authority.

CAK Connectivity Association Key.

CBC Cipher Block Chaining.

CCM Counter with CBC-MAC.

CKN Connectivity Association Key Name.

CMAC Cipher-based Message Authentication Code.

CMS Cryptographic Message Syntax.

CNSA Commercial National Security Algorithm Suite.

CPU Central Processing Unit.

CTR Counter Mode.

DH Diffie–Hellman.

DLP Discrete Logarithm Problem.

- DMA** Direct Memory Access.
- DNSSEC** Domain Name System Security Extensions.
- DPDK** Data Plane Development Kit.
- DPU** Data Processing Unit.
- ECC** Elliptic-Curve Cryptography.
- ECDH** Elliptic-Curve Diffie–Hellman.
- ECDLP** Elliptic-Curve Discrete Logarithm Problem.
- ECDSA** Elliptic-Curve Digital Signature Algorithm.
- EdDSA** Edwards-curve Digital Signature Algorithm.
- ENISA** European Union Agency for Cybersecurity.
- ESP** Encapsulating Security Payload.
- FIPS** Federal Information Processing Standards.
- FN-DSA** FFT over NTRU-Lattice-Based Digital Signature Algorithm.
- FO** Fujisaki–Okamoto Transformation.
- GCM** Galois/Counter Mode.
- GPU** Graphics Processing Unit.
- HKDF** HMAC-based Extract-and-Expand Key Derivation Function.
- HMAC** Hash-based Message Authentication Code.
- HNDL** Harvest Now, Decrypt Later.
- HPC** High-Performance Computing.
- HQC** Hamming Quasi-Cyclic.
- I/O** Input/Output.
- ICK** Integrity Check Key.
- ICV** Integrity Check Value.
- IETF** Internet Engineering Task Force.
- IFP** Integer Factorization Problem.

- IKEv2** Internet Key Exchange version 2.
- IKM** Input Key Material.
- IND-CCA** Indistinguishability under Chosen-Ciphertext Attack.
- IND-CPA** Indistinguishability under Chosen-Plaintext Attack.
- IP** Internet Protocol.
- IPsec** Internet Protocol Security.
- KDF** Key Derivation Function.
- KEK** Key Encryption Key.
- KEM** Key Encapsulation Mechanism.
- KEX** Key Exchange.
- KMS** Key Management System.
- KRI** Key Renewal Interval.
- LMS** Leighton-Micali Signature system.
- LWE** Learning With Errors.
- MAC** Message Authentication Code.
- MACsec** Media Access Control Security.
- MKA** MACsec Key Agreement.
- ML-DSA** Module-Lattice-Based Digital Signature Algorithm.
- ML-KEM** Module-Lattice-Based Key-Encapsulation Mechanism.
- MLWE** Module Learning With Errors.
- MPI** Message Passing Interface.
- MSK** Master Session Key.
- MTU** Maximum Transmission Unit.
- NIC** Network Interface Card.
- NIST** National Institute of Standards and Technology.
- NTT** Number-Theoretic Transform.

OQS Open Quantum Safe.

OSI Open Systems Interconnection.

OVS Open vSwitch.

PCIe Peripheral Component Interconnect Express.

PFS Perfect Forward Secrecy.

PKI Public Key Infrastructure.

PQC Post-Quantum Cryptography.

PRK Pseudorandom Key.

QKD Quantum Key Distribution.

RDMA Remote Direct Memory Access.

RPC Remote Procedure Call.

RSA Rivest–Shamir–Adleman.

S/MIME Secure/Multipurpose Internet Mail Extensions.

SAK Secure Association Key.

SecTAG Security Tag.

SEQ Sequence Number.

SHA-3 Secure Hash Algorithm 3.

SHAKE Secure Hash Algorithm KECCAK.

SIMD Single Instruction, Multiple Data.

SIS Short Integer Solution.

SLH-DSA Stateless Hash-Based Digital Signature Algorithm.

SmartNIC Smart Network Interface Card.

SPI Security Parameters Index.

SR-IOV Single Root I/O Virtualization.

SSH Secure Shell.

TCP Transmission Control Protocol.

TCP/IP Transmission Control Protocol/Internet Protocol.

TLS Transport Layer Security.

UDP User Datagram Protocol.

VF Virtual Function.

VFIO Virtual Function I/O.

VM Virtual Machine.

XFRM Transform Framework.

XMSS eXtended Merkle Signature Scheme.

XOF Extendable-Output Function.

Bibliography

- [1] A. C. Aguilera. “Accelerating Quantum-Secure Communications via DPU-Based Network Offloads”. PhD thesis. Eindhoven, The Netherlands: Eindhoven University of Technology, Feb. 2026. ISBN: 978-90-386-6598-6.
- [2] D. C. Lawo. “Post-Quantum Cryptography for Resilient and Secure Communication Systems”. PhD thesis. Eindhoven, The Netherlands: Eindhoven University of Technology, Dec. 2025. ISBN: 978-90-386-6546-7.
- [3] A. Cano Aguilera, X. Arnal i Clemente, D. C. Lawo, I. Tafur Monroy, and J. J. Vegas Olmos. “First End-to-End PQC Protected DPU-to-DPU Communications”. In: *Electronics Letters* 59.17 (2023), e12901. DOI: 10.1049/e112.12901.
- [4] D. Lawo, R. Frantz, A. Cano Aguilera, I. Tafur Monroy, and J. J. Vegas Olmos. “Implementation and Comparison of Dilithium/Kyber and Falcon/Kyber PQC Software Stack on Data Processing Units”. In: *Proceedings of the 27th Annual Symposium of the IEEE Photonics Benelux Chapter*. 27th Annual Symposium of the IEEE Photonics Benelux Chapter, IEEE Benelux 2023. Conference date: 23–24 November 2023. Ghent, Belgium, Nov. 2023. URL: <https://www.aanmelder.nl/ieee-ps-benelux-2023>.
- [5] A. Cano Aguilera, C. Rubio García, D. Lawo, R. Frantz, I. Tafur Monroy, J. Imaña, and J. Vegas Olmos. “Post-Quantum Secure Protocols for Communication Between Data Processing Units”. In: *Proceedings of IEEE Photonics Benelux Chapter Annual Symposium 2023*. 27th Annual Symposium of the IEEE Photonics Benelux Chapter, IEEE Benelux 2023. Conference date: 23–24 November 2023. Nov. 2023. URL: <https://www.aanmelder.nl/ieee-ps-benelux-2023>.
- [6] D. Lawo, R. Frantz, A. Cano Aguilera, X. Arnal I Clemente, M. Podleś, J. L. Imaña, I. Tafur Monroy, and J. J. Vegas Olmos. “Falcon/Kyber and Dilithium/Kyber Network Stack on NVIDIA’s Data Processing Unit Platform”. In: *IEEE Access* 12 (Mar. 2024), pp. 38048–38056. DOI: 10.1109/ACCESS.2024.3374629.
- [7] A. Cano Aguilera, R. Abu Bakar, F. Alhamed, C. Rubio Garcia, J. L. Imaña, I. Tafur Monroy, F. Cugini, and J. J. Vegas Olmos. “First Line-Rate End-to-End Post-Quantum Encrypted Optical Fiber Link Using Data Processing Units (DPUs)”.

- In: *Optical Fiber Communication Conference (OFC) 2024*. Optica Publishing Group, 2024, M1G.4. DOI: 10.1364/OFC.2024.M1G.4.
- [8] C. Rubio Garcia, A. Cano, J. J. Vegas Olmos, S. Rommel, and I. Tafur Monroy. “Integrating Quantum Key Distribution into TLS 1.3: A Transport Layer Approach to Quantum-Resistant Communications in Optical Networks”. In: *Optical Fiber Communication Conference (OFC) 2024*. Optica Publishing Group, 2024, Th3B.3. DOI: 10.1364/OFC.2024.Th3B.3.
- [9] A. Cano Aguilera, C. Rubio Garcia, D. Lawo, J. L. Imaña, I. Tafur Monroy, and J. J. Vegas Olmos. “In-line rate encrypted links using pre-shared post-quantum keys and DPUs”. In: *Scientific Reports* 14.1 (Sept. 2024), p. 21227. DOI: 10.1038/s41598-024-71861-x.
- [10] D. C. Lawo, R. Abu Bakar, A. Cano Aguilera, F. Cugini, J. L. Imaña, I. Tafur Monroy, and J. J. Vegas Olmos. “Wireless and Fiber-Based Post-Quantum-Cryptography-Secured IPsec Tunnel”. In: *Future Internet* 16.8 (2024), p. 300. DOI: 10.3390/fi16080300.
- [11] A. Cano, C. R. Garcia, R. Frantz, I. T. Monroy, J. L. Imaña, and J. J. Vegas. “Integrating Post-Quantum Cryptography Plugins for IPsec Offloads to Data Processing Units in the Cloud-Edge Continuum”. In: *2024 IEEE 32nd International Conference on Network Protocols (ICNP)*. IEEE, 2024, pp. 1–6. DOI: 10.1109/ICNP61940.2024.10858568.
- [12] D. Lawo, M. Podleś, R. Frantz, A. Cano Aguilera, D. Iliadis-Apostolidis, J. Sánchez García, S. Kosta, I. Tafur Monroy, J. L. Imaña, and J. J. Vegas Olmos. “The Zero-Tax Data Center: A Use Case Through Quantum Resilient Communications”. In: *2024 24th International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2024, pp. 1–4. DOI: 10.1109/ICTON62926.2024.10647337.
- [13] C. Rubio García, A. Cano Aguilera, C. Stan, J. J. Vegas Olmos, S. Rommel, and I. Tafur Monroy. “Enhanced Network Security Protocols for the Quantum Era: Combining Classical and Post-Quantum Cryptography, and Quantum Key Distribution”. In: *IEEE Journal on Selected Areas in Communications* 43.8 (2025), pp. 2765–2781. DOI: 10.1109/JSAC.2025.3568011.
- [14] A. Cano Aguilera, C. Rubio Garcia, D. C. Lawo, I. Tafur, J. L. Imaña, and J. J. Vegas Olmos. “First Demonstration of 200 Gbps Regime Line-Rate Quantum-Secure MACsec Optical Links Using Commodity Hardware Offloads”. In: *Optical Fiber Communication Conference (OFC) 2025*. Optica Publishing Group, 2025, Tu2D.2. DOI: 10.1364/OFC.2025.Tu2D.2.
- [15] A. Cano Aguilera, I. Tafur Monroy, J. J. Vegas Olmos, and J. L. Imaña. “ARM Architecture Optimizations for Line-Rate PQC Communications”. In: *2025 International Conference on Optical Network Design and Modeling (ONDM)*. IEEE, 2025, pp. 1–4. DOI: 10.23919/ONDM65745.2025.11029331.

- [16] A. Cano, J. L. Imaña, I. T. Monroy, and J. J. Vegas Olmos. “Post-Quantum Cryptography for Quantum Resilient Data Transfer”. In: *2025 25th Anniversary International Conference on Transparent Optical Networks (ICTON)*. Ed. by C. Cojocaru, S. Spadaro, and M. Marciniak. United States: Institute of Electrical and Electronics Engineers, Aug. 2025. DOI: 10.1109/ICTON67126.2025.11125195.
- [17] D. Lawo, R. Abu Bakar, A. Cano Aguilera, F. Cugini, J. L. Imaña, I. Tafur Monroy, and J. J. Vegas Olmos. “Future Data Centers: Hardware-Offloaded Post-Quantum Secure Optical IPsec Tunnel for Confidential AI Training”. In: *Proceedings of the 2025 International Conference on Optical Network Design and Modeling (ONDM)*. Pisa, Italy, 2025. URL: <https://opendl.ifip-tc6.org/db/conf/ondm2025/ondm2025/1571114929.pdf>.
- [18] D. Iliadis-Apostolidis, D. C. Lawo, S. Kosta, I. Tafur Monroy, and J. J. Vegas Olmos. “QRoNS: Quantum Resilience Over IPsec Tunnels for Network Slicing”. In: *Electronics* 14.21 (2025), p. 4234. DOI: 10.3390/electronics14214234.
- [19] D. Iliadis-Apostolidis, D. C. Lawo, S. Kosta, and J. J. Vegas Olmos. “SNSVS: Scalable Network Slicing with Virtualized Systems using Post-Quantum Cryptography”. In: *GLOBECOM 2025 - 2025 IEEE Global Communications Conference*. IEEE, 2025, pp. 3109–3114. DOI: 10.1109/GLOBECOM59602.2025.11431714.
- [20] D. C. Lawo, I. T. Monroy, and J. J. V. Olmos. “Confidential Computing and AI in Networking Fabrics”. In: *2025 25th Anniversary International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2025, pp. 1–4. DOI: 10.1109/ICTON67126.2025.11125026.
- [21] J. Meyer, A. Cano, F. Cugini, I. Tafur Monroy, and J. J. Vegas Olmos. *Design and Evaluation of a Heterogeneous DPU Architecture for Accelerating Post-Quantum Cryptography*. Preprint, version 1, Research Square. Nov. 2025. DOI: 10.21203/rs.3.rs-7817696/v1.
- [22] P. W. Shor. “Algorithms for Quantum Computation: Discrete Logarithms and Factoring”. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. IEEE, 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.
- [23] S. Kanev, J. P. Darago, K. Hazelwood, P. Ranganathan, T. Moseley, G.-Y. Wei, and D. Brooks. “Profiling a Warehouse-Scale Computer”. In: *Proceedings of the 42nd Annual International Symposium on Computer Architecture*. ISCA ’15. Association for Computing Machinery, 2015, pp. 158–169. DOI: 10.1145/2749469.2750392.
- [24] S. Kanev, J. P. Darago, K. Hazelwood, P. Ranganathan, T. Moseley, G.-Y. Wei, and D. Brooks. “Retrospective: Profiling a Warehouse-Scale Computer”. In: *International Symposium on Computer Architecture (ISCA)*. ISCA@50 25-Year Retrospective: 1996–2020. 2023. URL: https://skanev.org/papers/wsc_retrospective.pdf.

- [25] IEEE Computer Society LAN/MAN Standards Committee. *IEEE P802.3dj Project Authorization Request: Standard for Ethernet Amendment: Media Access Control Parameters for 1.6 Tb/s and Physical Layers and Management Parameters for 200 Gb/s, 400 Gb/s, 800 Gb/s, and 1.6 Tb/s Operation*. Project Authorization Request P802.3dj. PAR approved 3 December 2022. Project status active at time of citation. IEEE, Sept. 2022. URL: https://www.ieee802.org/3/dj/projdoc/P802d3dj_PAR.pdf.
- [26] C. Gidney. *How to Factor 2048 Bit RSA Integers with Less Than a Million Noisy Qubits*. 2025. DOI: 10.48550/arXiv.2505.15917. arXiv: 2505.15917 [quant-ph].
- [27] Intro to Quantum. *The Timelines: When Can We Expect Useful Quantum Computers?* Intro to Quantum. Accessed 25 August 2026. 2024. URL: <https://introtoquantum.org/essentials/timelines/>.
- [28] C. Gidney and M. Ekerå. “How to Factor 2048 Bit RSA Integers in 8 Hours Using 20 Million Noisy Qubits”. In: *Quantum* 5 (Apr. 2021), p. 433. DOI: 10.22331/q-2021-04-15-433.
- [29] J. Mascelli and M. Rodden. “*Harvest Now Decrypt Later*”: *Examining Post-Quantum Cryptography and the Data Privacy Risks for Distributed Ledger Networks*. Finance and Economics Discussion Series 2025-093. Board of Governors of the Federal Reserve System, Sept. 2025. DOI: 10.17016/FEDS.2025.093.
- [30] General Intelligence and Security Service of the Netherlands (AIVD), Centrum Wiskunde & Informatica (CWI), and TNO. *The PQC Migration Handbook: Guidelines for Migrating to Post-Quantum Cryptography*. Guidance Report. Second edition. Publication date: 29 November 2024. General Intelligence and Security Service of the Netherlands (AIVD), Nov. 2024. URL: <https://english.aivd.nl/documents/2024/12/3/the-pqc-migration-handbook>.
- [31] Cybersecurity Authorities of 18 European Union Member States. *Securing Tomorrow, Today: Transitioning to Post-Quantum Cryptography*. Joint Statement. Joint statement from partners from 18 European Union member states. Federal Office for Information Security (BSI), Nov. 2024. URL: <https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Crypto/PQC-joint-statement.html>.
- [32] National Security Agency. *The Commercial National Security Algorithm Suite 2.0 and Quantum Computing FAQ*. Cybersecurity Information Sheet U/OO/194427-22. Version 2.1; PP-24-4014. National Security Agency, Dec. 2024. URL: https://media.defense.gov/2022/Sep/07/2003071836/-1/-1/0/CSI_CNNSA_2.0_FAQ_.PDF.
- [33] Agence nationale de la sécurité des systèmes d’information (ANSSI). *ANSSI Views on Crypto Agility*. Technical Guidance. Published 19 January 2026. ANSSI, Jan. 2026. URL: <https://messervices.cyber.gouv.fr/guides/ANSSI-views-on-crypto-agility>.

- [34] D. J. Bernstein. “Introduction to Post-Quantum Cryptography”. In: *Post-Quantum Cryptography*. Ed. by D. J. Bernstein, J. Buchmann, and E. Dahmen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1–14. DOI: 10.1007/978-3-540-88702-7_1.
- [35] G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, Y.-K. Liu, C. Miller, D. Moody, et al. *Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process*. NIST Internal Report 8413-upd1. National Institute of Standards and Technology, Sept. 2022. DOI: 10.6028/NIST.IR.8413-upd1.
- [36] National Institute of Standards and Technology. *Module-Lattice-Based Key-Encapsulation Mechanism Standard*. Federal Information Processing Standards Publication NIST FIPS 203. Washington, D.C.: U.S. Department of Commerce, Aug. 2024. DOI: 10.6028/NIST.FIPS.203.
- [37] National Institute of Standards and Technology. *Module-Lattice-Based Digital Signature Standard*. Federal Information Processing Standards Publication NIST FIPS 204. Washington, D.C.: U.S. Department of Commerce, Aug. 2024. DOI: 10.6028/NIST.FIPS.204.
- [38] National Institute of Standards and Technology. *Stateless Hash-Based Digital Signature Standard*. Federal Information Processing Standards Publication NIST FIPS 205. Washington, D.C.: U.S. Department of Commerce, Aug. 2024. DOI: 10.6028/NIST.FIPS.205.
- [39] F. Xu, X. Ma, Q. Zhang, H.-K. Lo, and J.-W. Pan. “Secure Quantum Key Distribution With Realistic Devices”. In: *Reviews of Modern Physics* 92.2 (May 2020), p. 025002. DOI: 10.1103/RevModPhys.92.025002.
- [40] R. Rios and J. A. Montenegro. “Post-Quantum Cryptography: A Multi-layer Bottleneck Analysis”. In: *International Journal of Information Security* 25.3, 89 (May 2026). DOI: 10.1007/s10207-026-01242-0.
- [41] D. Boneh and V. Shoup. *A Graduate Course in Applied Cryptography*. Online textbook. Version 0.6; accessed 25 August 2026. 2023. URL: <https://toc.cryptobook.us/>.
- [42] C. Paar and J. Pelzl. *Understanding Cryptography: A Textbook for Students and Practitioners*. Berlin, Heidelberg: Springer, 2010. DOI: 10.1007/978-3-642-04101-3.
- [43] C. Matt and U. Maurer. “The One-Time Pad Revisited”. In: *2013 IEEE International Symposium on Information Theory*. IEEE, 2013, pp. 2706–2710. DOI: 10.1109/ISIT.2013.6620718.

- [44] National Institute of Standards and Technology. *Advanced Encryption Standard (AES)*. Federal Information Processing Standards Publication NIST FIPS 197-upd1. Updated version of FIPS 197, originally published in 2001. Washington, D.C.: U.S. Department of Commerce, May 2023. DOI: 10.6028/NIST.FIPS.197-upd1.
- [45] J. Daemen and V. Rijmen. “The Block Cipher Rijndael”. In: *Smart Card Research and Applications*. Ed. by J.-J. Quisquater and B. Schneier. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 277–284. DOI: 10.1007/10721064_26.
- [46] M. J. Dworkin. *Recommendation for Block Cipher Modes of Operation: Methods and Techniques*. NIST Special Publication 800-38A. Gaithersburg, MD: National Institute of Standards and Technology, Dec. 2001. DOI: 10.6028/NIST.SP.800-38A.
- [47] M. J. Dworkin. *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*. NIST Special Publication 800-38D. Gaithersburg, MD: National Institute of Standards and Technology, Nov. 2007. DOI: 10.6028/NIST.SP.800-38D.
- [48] H. Krawczyk, M. Bellare, and R. Canetti. *HMAC: Keyed-Hashing for Message Authentication*. RFC 2104. RFC Editor, Feb. 1997. DOI: 10.17487/RFC2104.
- [49] M. J. Dworkin. *Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication*. NIST Special Publication 800-38B. Gaithersburg, MD: National Institute of Standards and Technology, Oct. 2016. DOI: 10.6028/NIST.SP.800-38B.
- [50] E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. RFC Editor, Aug. 2018. DOI: 10.17487/RFC8446.
- [51] J. Viega and D. McGrew. *The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)*. RFC 4106. RFC Editor, June 2005. DOI: 10.17487/RFC4106.
- [52] IEEE. *IEEE Standard for Local and Metropolitan Area Networks—Media Access Control (MAC) Security*. IEEE Std 802.1AE-2018. IEEE, Dec. 2018. DOI: 10.1109/IEEESTD.2018.8585421.
- [53] H. Krawczyk and P. Eronen. *HMAC-based Extract-and-Expand Key Derivation Function (HKDF)*. RFC 5869. RFC Editor, May 2010. DOI: 10.17487/RFC5869.
- [54] National Institute of Standards and Technology. *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*. Federal Information Processing Standards Publication NIST FIPS 202. Gaithersburg, MD: U.S. Department of Commerce, Aug. 2015. DOI: 10.6028/NIST.FIPS.202.
- [55] Certicom Research. *SEC 1: Elliptic Curve Cryptography*. Standard SEC 1, Version 2.0. Standards for Efficient Cryptography Group, May 2009. URL: <https://www.secg.org/sec1-v2.pdf>.

- [56] W. Diffie and M. Hellman. “New Directions in Cryptography”. In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654. DOI: 10.1109/TIT.1976.1055638.
- [57] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC 5280. RFC Editor, May 2008. DOI: 10.17487/RFC5280.
- [58] International Telecommunication Union. *Information technology — Open Systems Interconnection — The Directory: Public-key and attribute certificate frameworks*. Recommendation X.509. Geneva, Switzerland: ITU-T, 2019. URL: <https://handle.itu.int/11.1002/1000/14033>.
- [59] K. S. Mohamed. “Cryptography Concepts: Integrity, Authentication, Availability, Access Control, and Non-repudiation”. In: *New Frontiers in Cryptography: Quantum, Blockchain, Lightweight, Chaotic and DNA*. Cham: Springer Cham, 2020, pp. 41–63. DOI: 10.1007/978-3-030-58996-7_3.
- [60] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, and T. Kivinen. *Internet Key Exchange Protocol Version 2 (IKEv2)*. RFC 7296. RFC Editor, Oct. 2014. DOI: 10.17487/RFC7296.
- [61] S. Kent. *IP Encapsulating Security Payload (ESP)*. RFC 4303. RFC Editor, Dec. 2005. DOI: 10.17487/RFC4303.
- [62] S. Kent and K. Seo. *Security Architecture for the Internet Protocol*. RFC 4301. RFC Editor, Dec. 2005, p. 101. DOI: 10.17487/RFC4301.
- [63] S. Dubroca. “MACsec: Encryption for the Wired LAN”. In: *Proceedings of Netdev 1.1*. 2016. URL: <https://netdevconf.info/1.1/proceedings/papers/MACsec-Encryption-for-the-wired-LAN.pdf>.
- [64] IEEE. *IEEE Standard for Local and Metropolitan Area Networks—Port-Based Network Access Control*. IEEE Std 802.1X-2010. Revision of IEEE Std 802.1X-2004. IEEE, 2010, pp. 1–205. DOI: 10.1109/IEEESTD.2010.5409813.
- [65] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, et al. “Quantum Supremacy Using a Programmable Superconducting Processor”. In: *Nature* 574.7779 (Oct. 2019), pp. 505–510. DOI: 10.1038/s41586-019-1666-5.
- [66] V. Bhatia and K. R. Ramkumar. “An Efficient Quantum Computing Technique for Cracking RSA Using Shor’s Algorithm”. In: *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*. IEEE, 2020, pp. 89–94. DOI: 10.1109/ICCCA49541.2020.9250806.

- [67] Network and Information Systems Cooperation Group. *A Coordinated Implementation Roadmap for the Transition to Post-Quantum Cryptography*. Implementation Roadmap. European Commission, June 2025. URL: <https://digital-strategy.ec.europa.eu/en/library/coordinated-implementation-roadmap-transition-post-quantum-cryptography>.
- [68] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki. “Quantum Entanglement”. In: *Reviews of Modern Physics* 81.2 (June 2009), pp. 865–942. DOI: 10.1103/RevModPhys.81.865.
- [69] J. R. Friedman, V. Patel, W. Chen, S. K. Tolpygo, and J. E. Lukens. “Quantum Superposition of Distinct Macroscopic States”. In: *Nature* 406.6791 (July 2000), pp. 43–46. DOI: 10.1038/35017505.
- [70] W. Easttom. “Quantum Computing and Cryptography”. In: *Modern Cryptography: Applied Mathematics for Encryption and Information Security*. Cham: Springer Cham, 2021, pp. 385–390. DOI: 10.1007/978-3-030-63115-4.
- [71] L. K. Grover. “A Fast Quantum Mechanical Algorithm for Database Search”. In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC ’96. Association for Computing Machinery, 1996, pp. 212–219. DOI: 10.1145/237814.237866.
- [72] X. Bonnetain, M. Naya-Plasencia, and A. Schrottenloher. “Quantum Security Analysis of AES”. In: *IACR Transactions on Symmetric Cryptology* 2019.2 (June 2019), pp. 55–93. DOI: 10.13154/tosc.v2019.i2.55-93.
- [73] M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt. “Applying Grover’s Algorithm to AES: Quantum Resource Estimates”. In: *Proceedings of the 7th International Workshop on Post-Quantum Cryptography - Volume 9606*. PQCrypto 2016. Springer-Verlag, 2016, pp. 29–43. DOI: 10.1007/978-3-319-29360-8_3.
- [74] G. Brassard, P. Høyer, and A. Tapp. “Quantum Cryptanalysis of Hash and Claw-Free Functions”. In: *LATIN’98: Theoretical Informatics*. Ed. by C. L. Lucchesi and A. V. Moura. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 163–169. DOI: 10.1007/BFb0054319.
- [75] D. Moody, R. Perlner, A. Regenscheid, A. Robinson, and D. Cooper. *Transition to Post-Quantum Cryptography Standards*. NIST Internal Report 8547 ipd. National Institute of Standards and Technology, Nov. 2024. DOI: 10.6028/NIST.IR.8547.ipd.
- [76] M. Mosca and M. Piani. *Quantum Threat Timeline Report 2024*. Research Report. Report by Global Risk Institute and evolutionQ. Global Risk Institute, Dec. 2024. URL: <https://globalriskinstitute.org/publication/2024-quantum-threat-timeline-report/>.

- [77] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle. “CRYSTALS-Kyber: A CCA-Secure Module-Lattice-Based KEM”. In: *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. 2018, pp. 353–367. DOI: 10.1109/EuroSP.2018.00032.
- [78] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. “CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme”. In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2018.1 (Feb. 2018), pp. 238–268. DOI: 10.13154/tches.v2018.i1.238-268.
- [79] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe. “The SPHINCS+ Signature Framework”. In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 2129–2146. DOI: 10.1145/3319535.3363229.
- [80] C. Aguilar-Melchor, O. Blazy, J.-C. Deneuville, P. Gaborit, and G. Zémor. “Efficient Encryption From Random Quasi-Cyclic Codes”. In: *IEEE Transactions on Information Theory* 64.5 (2018), pp. 3927–3943. DOI: 10.1109/TIT.2018.2804444.
- [81] A. Kipnis and A. Shamir. “Cryptanalysis of the Oil and Vinegar Signature Scheme”. In: *Advances in Cryptology – CRYPTO ’98*. Ed. by H. Krawczyk. Vol. 1462. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 257–266. DOI: 10.1007/BFb0055733.
- [82] W. Beullens. “Breaking Rainbow Takes a Weekend on a Laptop”. In: *Advances in Cryptology – CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part II*. Berlin, Heidelberg: Springer-Verlag, 2022, pp. 464–479. DOI: 10.1007/978-3-031-15979-4_16.
- [83] D. A. Cooper, D. C. Apon, Q. H. Dang, M. S. Davidson, M. J. Dworkin, and C. A. Miller. *Recommendation for Stateful Hash-Based Signature Schemes*. NIST Special Publication 800-208. Gaithersburg, MD: National Institute of Standards and Technology, Oct. 2020. DOI: 10.6028/NIST.SP.800-208.
- [84] G. Alagic, M. Bros, P. Ciadoux, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, Y.-K. Liu, C. Miller, et al. *Status Report on the Fourth Round of the NIST Post-Quantum Cryptography Standardization Process*. NIST Internal Report 8545. Gaithersburg, MD: National Institute of Standards and Technology, Mar. 2025. DOI: 10.6028/NIST.IR.8545.
- [85] M. O. Alshomrany, F. Alsolami, M. Al-Hashimi, M. Saleh, O. Abulnaja, T. F. Al-Somani, and A. Altuwaijri. “QUASAR: Achieving Quantum Readiness for Post-Quantum Cryptography on RISC-V at Minimum Hardware Cost”. In: *Electronics* 15.10 (May 2026), p. 2154. DOI: 10.3390/electronics15102154.

- [86] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang. *Falcon: Fast-Fourier Lattice-Based Compact Signatures over NTRU*. Specification, version 1.2, submitted to the NIST Post-Quantum Cryptography Standardization Process. Jan. 2020. URL: <https://falcon-sign.info/falcon.pdf>.
- [87] J. Howe, T. Prest, and D. Apon. “SoK: How (not) to Design and Implement Post-Quantum Cryptography”. In: *Topics in Cryptology – CT-RSA 2021*. Ed. by K. G. Paterson. Vol. 12704. Lecture Notes in Computer Science. Cham: Springer, 2021, pp. 444–477. DOI: 10.1007/978-3-030-75539-3_19.
- [88] D. J. Bernstein, K. Bhargavan, S. Bhasin, A. Chattopadhyay, T. K. Chia, M. J. Kannwischer, F. Kiefer, T. B. Paiva, P. Ravi, and G. Tamvada. “KyberSlash: Exploiting Secret-Dependent Division Timings in Kyber Implementations”. In: *IACR Transactions on Cryptographic Hardware and Embedded Systems 2025.2* (Mar. 2025), pp. 209–234. DOI: 10.46586/tches.v2025.i2.209-234.
- [89] National Vulnerability Database. *CVE-2024-37880 Detail*. National Vulnerability Database. Accessed 25 August 2026. 2024. URL: <https://nvd.nist.gov/vuln/detail/CVE-2024-37880>.
- [90] R. Rios, J. A. Montenegro, A. Muñoz, and D. Ferraris. “Toward the Quantum-Safe Web: Benchmarking Post-Quantum TLS”. In: *IEEE Network* 39.5 (Sept. 2025), pp. 247–253. DOI: 10.1109/MNET.2025.3531116.
- [91] D. P. Karcz, M. Niemiec, M. A. Kourtis, and T. Köppl. “Exploring Hardware Implementation Feasibility of Post-Quantum Cryptography in Embedded Systems: Evaluation of NIST-Standardized ML-KEM, ML-DSA, and SLH-DSA on ESP32-C6”. In: *PLOS ONE* 21.8 (Aug. 2026), e0355179. DOI: 10.1371/journal.pone.0355179.
- [92] O. Gillot, W. J. Buchanan, and M. G. Tehrani. “Energy Consumption of Post-Quantum Cryptography on Constrained and General-Purpose Architectures”. In: *Cryptography* 10.4 (2026), p. 55. DOI: 10.3390/cryptography10040055.
- [93] European Union Agency for Cybersecurity (ENISA). *Hybridization of Traditional Cryptographic Mechanisms with PQC: Standardisation Status*. Standardisation Status Report. Published 30 April 2026; this report does not establish an ENISA/ECCG recommendation. European Union Agency for Cybersecurity (ENISA), Apr. 2026. URL: https://certification.enisa.europa.eu/document/download/02f54596-6a99-4c9e-88c4-8c711ebff9c1_en?filename=ENISA+report+-+Hybridization+of+traditional+cryptographic+mechanisms+with+PQC+-+Standardisation+Status+-+30+April+2026+.pdf.
- [94] J. Haridas and M. Bachman. *PQC in Plaintext: Google Cloud’s Post-Quantum Cryptography Roadmap*. Google Cloud Blog. Accessed 25 August 2026. Aug. 2026. URL: <https://cloud.google.com/blog/products/identity-security/pqc-in-plaintext-google-clouds-post-quantum-cryptography-roadmap>.

- [95] Apple Inc. *Prepare Your Network for Quantum-Secure Encryption in TLS*. Apple Support. Accessed 25 August 2026. June 2026. URL: <https://support.apple.com/en-la/122756>.
- [96] A. Weibel. *ML-KEM Post-Quantum TLS Now Supported in AWS KMS, ACM, and Secrets Manager*. AWS Security Blog. Accessed 25 August 2026. Apr. 2025. URL: <https://aws.amazon.com/blogs/security/ml-kem-post-quantum-tls-now-supported-in-aws-kms-acm-and-secrets-manager/>.
- [97] T. Daffron. *Future-Proofing the Internet: Akamai Achieves End-to-End PQC*. Akamai Blog. Accessed 25 August 2026. Aug. 2026. URL: <https://www.akamai.com/blog/security/future-proofing-internet-akamai-achieves-end-to-end-pqc>.
- [98] European Cybersecurity Certification Group, Sub-group on Cryptography. *Agreed Cryptographic Mechanisms*. Cryptographic Guidance. Version 2.0; applicable from 6 May 2025. European Cybersecurity Certification Group, Apr. 2025. URL: https://certification.enisa.europa.eu/document/download/a845662b-ae0-484e-9191-890c4cfa7aaa_en?filename=ECCG+Agreed+Cryptographic+Mechanisms+version+2.pdf.
- [99] Federal Office for Information Security (BSI). *Cryptographic Mechanisms: Recommendations and Key Lengths – Part 2: Use of Transport Layer Security (TLS)*. Technical Guideline BSI TR-02102-2. Version 2025-1. Federal Office for Information Security (BSI), 2025. URL: <https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102-2.html>.
- [100] Federal Office for Information Security (BSI). *Cryptographic Mechanisms: Recommendations and Key Lengths – Part 3: Use of Internet Protocol Security (IPsec) and Internet Key Exchange (IKEv2)*. Technical Guideline BSI TR-02102-3. Version 2025-01. Federal Office for Information Security (BSI), 2025. URL: <https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102-3.html>.
- [101] Agence nationale de la sécurité des systèmes d’information (ANSSI). *Cryptographie post-quantique (PQC)*. Official ANSSI guidance portal. Accessed 24 August 2026. URL: <https://cyber.gouv.fr/enjeux-technologiques/cryptographie-post-quantique>.
- [102] C. Tjhai, M. Tomlinson, G. Bartlett, S. Fluhrer, D. Van Geest, O. Garcia-Morchon, and V. Smyslov. *Multiple Key Exchanges in the Internet Key Exchange Protocol Version 2 (IKEv2)*. RFC 9370. RFC Editor, May 2023. DOI: 10.17487/RFC9370.
- [103] S. Fluhrer, P. Kampanakis, D. McGrew, and V. Smyslov. *Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security*. RFC 8784. RFC Editor, June 2020. DOI: 10.17487/RFC8784.

- [104] S. J. Park, R. Govindan, K. Shen, D. Culler, F. Özcan, G.-W. Kim, and H. Levy. “Lovelock: Towards Smart NIC-Hosted Clusters”. In: *ACM SIGENERGY Energy Informatics Review* 4.5 (Apr. 2025), pp. 172–179. DOI: 10.1145/3727200.3727226.
- [105] A. Stephan and L. Wüstrich. “The Path of a Packet Through the Linux Kernel”. In: *Seminar IITM WS 23*. Technical University of Munich, Chair of Network Architectures and Services, 2024, pp. 89–93. DOI: 10.2313/NET-2024-04-1_16.
- [106] M. Hossain. *Trends in Data Center Security: Part 1: Traffic Trends*. Cisco Security Blog. Accessed 25 August 2026. May 2014. URL: <https://blogs.cisco.com/security/trends-in-data-center-security-part-1-traffic-trends>.
- [107] N. Kamiyama, Y. Nakano, K. Shiimoto, G. Hasegawa, M. Murata, and H. Miyahara. “Investigating Structure of Modern Web Traffic”. In: *2015 IEEE 16th International Conference on High Performance Switching and Routing (HPSR)*. IEEE, 2015, pp. 1–8. DOI: 10.1109/HPSR.2015.7483102.
- [108] Intel Corporation. *Intel Ethernet Adapters and Devices User Guide: Single Root I/O Virtualization (SR-IOV)*. Document ID: 705831. Version 29.3. Intel Corporation. Nov. 2024. URL: <https://edc.intel.com/content/www/us/en/design/products/ethernet/adapters-and-devices-user-guide/29.3/single-root-i-o-virtualization-sr-iov/>.
- [109] V. Del Piccolo, A. Amamou, K. Haddadou, and G. Pujolle. “A Survey of Network Isolation Solutions for Multi-Tenant Data Centers”. In: *IEEE Communications Surveys & Tutorials* 18.4 (2016), pp. 2787–2821. DOI: 10.1109/COMST.2016.2556979.
- [110] V. G. Cerf and E. Cain. “The DoD Internet Architecture Model”. In: *Computer Networks (1976)* 7.5 (1983), pp. 307–318. DOI: 10.1016/0376-5075(83)90042-9.
- [111] International Organization for Standardization and International Electrotechnical Commission. *Information Technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*. International Standard ISO/IEC 7498-1:1994. ISO/IEC, Nov. 1994. URL: <https://www.iso.org/standard/20269.html>.
- [112] S. Goldberg and A. Paul. *Post-Quantum Encryption for Cloudflare IPsec Is Generally Available*. Cloudflare Blog. Accessed 25 August 2026. Apr. 2026. URL: <https://blog.cloudflare.com/post-quantum-ipsec/>.
- [113] Apple Inc. *Quantum-Secure Cryptography in Apple Operating Systems*. Apple Platform Security. Apple Inc. Jan. 2026. URL: <https://support.apple.com/guide/security/quantum-secure-cryptography-apple-devices-secc7c82e533/web>.
- [114] Cisco Systems, Inc. *Post-Quantum Cryptography on Cisco 8000 Series Secure Routers for IKEv2 Sessions*. Security and VPN Configuration Guide. Cisco Systems, Inc. Apr. 2026. URL: <https://www.cisco.com/c/en/us/td/docs/routers/ios-xe/security-vpn/security-vpn/m-pqc-ikev2.html>.

- [115] M. Perera, M. Mackay, M. Hashem Eiza, A. Raschella, N. Shone, and M. K. Maheshwari. “Towards Quantum-Safe O-RAN: Experimental Evaluation of ML-KEM-Based IPsec on the E2 Interface”. In: *Future Internet* 18.4 (Apr. 2026), p. 188. DOI: 10.3390/fi18040188.
- [116] P. Kampanakis. *Post-Quantum Key Exchange with ML-KEM in the Internet Key Exchange Protocol Version 2 (IKEv2)*. Internet-Draft draft-ietf-ipsecme-ikev2-mlkem-09. Work in Progress. Internet Engineering Task Force, July 2026. URL: <https://datatracker.ietf.org/doc/draft-ietf-ipsecme-ikev2-mlkem/09/>.
- [117] B. Westerbaan. *State of the Post-Quantum Internet in 2025*. Cloudflare Blog. Accessed 25 August 2026. Oct. 2025. URL: <https://blog.cloudflare.com/pq-2025/>.
- [118] E. C. Santos, T. A. O. D. Cordeiro, and H. de Oliveira Silva. “Performance Evaluation of Post-Quantum Cryptography in IoT: A Case Study on MQTT Over TLS Under Network Constraints”. In: *2026 14th International Symposium on Digital Forensics and Security (ISDFS)*. IEEE, 2026, pp. 1–6. DOI: 10.1109/ISDFS69419.2026.11458974.
- [119] D. Adrian, B. Beck, D. Benjamin, and D. O’Brien. *Advancing Our Amazing Bet on Asymmetric Cryptography*. Chromium Blog. Accessed 25 August 2026. May 2024. URL: <https://blog.chromium.org/2024/05/advancing-our-amazing-bet-on-asymmetric.html>.
- [120] E. Ozturk, D. Zimmerman, K. Yap, M. Cornu, and T. Kantecki. *Post-Quantum Cryptography: Accelerating Open Quantum Safe Library with Intel AVX-512 Keccak 1600 Implementation*. Technology Guide. Last updated 4 November 2025. Intel Corporation, Nov. 2025. URL: <https://builders.intel.com/solutionslibrary/post-quantum-cryptography-accelerating-open-quantum-safe-library-with-intel-avx-512-keccak-1600-implementation>.
- [121] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee. “Network Function Virtualization: Challenges and Opportunities for Innovations”. In: *IEEE Communications Magazine* 53.2 (2015), pp. 90–97. DOI: 10.1109/MCOM.2015.7045396.
- [122] ETSI Industry Specification Group for Network Functions Virtualisation. *Network Functions Virtualisation (NFV) Release 3; Evolution and Ecosystem; Report on Network Slicing Support with ETSI NFV Architecture Framework*. ETSI Group Report ETSI GR NFV-EVE 012 V3.1.1. European Telecommunications Standards Institute, Dec. 2017. URL: https://www.etsi.org/deliver/etsi_gr/NFV-EVE/001_099/012/03.01.01_60/gr_nfv-eve012v030101p.pdf.
- [123] S. Kumar. *SR-IOV*. NgKore Documentation. Last updated 16 February 2026; accessed 25 August 2026. May 2023. URL: <https://docs.ngkorefoundation.org/kernel-bypass/sriov/>.

- [124] A. Banerjee, T. Reddy.K, D. Schoinianakis, T. Hollebeek, and M. Ounsworth. *Post-Quantum Cryptography for Engineers*. RFC 9958. RFC Editor, June 2026. DOI: 10.17487/RFC9958.
- [125] J. Stenstam. *Algorithm-Split DNSSEC: KSK/ZSK Algorithm Separation*. Internet-Draft draft-johani-dnsop-dnssec-alg-split-02. Work in Progress. Internet Engineering Task Force, July 2026. URL: <https://datatracker.ietf.org/doc/draft-johani-dnsop-dnssec-alg-split/02/>.
- [126] OpenSSH Project. *OpenSSH 10.0 Release Notes*. OpenSSH Release Notes. Accessed 25 August 2026. Apr. 2025. URL: <https://www.openssh.org/txt/release-10.0>.
- [127] J. Schaad, B. Ramsdell, and S. Turner. *Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification*. RFC 8551. RFC Editor, Apr. 2019. DOI: 10.17487/RFC8551.
- [128] B. Salter, A. Raine, and D. Van Geest. *Use of the ML-DSA Signature Algorithm in the Cryptographic Message Syntax (CMS)*. RFC 9882. RFC Editor, Oct. 2025. DOI: 10.17487/RFC9882.
- [129] J. Prat, M. Ounsworth, and D. Van Geest. *Use of ML-KEM in the Cryptographic Message Syntax (CMS)*. RFC 9936. RFC Editor, Mar. 2026. DOI: 10.17487/RFC9936.